UNCLASSIFIED

AD NUMBER
ADB027763
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; MAY 1978. Other requests shall be referred to Rome Air Development Center, Attn: IRDT, Griffiss AFB, NY 13441.**
AUTHORITY
RADC, USAF ltr, 26 Sep 1980

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED AND CLEARED FOR PUBLIC RELEASE UNDER DOD DIRECTIVE 5200.20 AND NO RESTRICTIONS ARE IMPOSED UPON ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

FOR FURTHER TRAN

AD B027763

RADC-TR-78-100
Final Technical Report
May 1978

LINGUISTIC DOCUMENTATION OF METAL SYSTEM

Winfred P. Lehmann
Solveig M. Pflueger
Helen-Jo J. Hewitt
Robert A. Amsler
Howard R. Smith

Distribution limited to U. S. Government agencies only; test and evaluation; May 1978. Other requests for this document must be referred to RADC (IRDT), Griffiss AFB NY 13441.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

RADC-TR-78-100 has been reviewed and is approved for publication.

APPROVED:

ZBIGNIEW L. PANKOWICZ
Project Engineer

APPROVED:

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRDT) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-78-100
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): LINGUISTIC DOCUMENTATION OF METAL SYSTEM
5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, 11 Apr 77 – 10 Jan 78
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Winfred P. Lehmann, Robert A. Amsler, Solveig M. Pflueger, Howard R. Smith, Helen-Jo J. Hewitt
8. CONTRACT OR GRANT NUMBER(s): F30602-77-C-0047
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The University of Texas at Austin, Linguistics Research Center, Austin TX 78712
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 45940829
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss AFB NY 13441
12. REPORT DATE: May 1978
13. NUMBER OF PAGES: 160 (171 p.)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same (4514) (92)
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U.S. Government agencies only; test and evaluation, May 1978. Other requests for this document must be referred to RADC (IRDT) Griffiss AFB NY 13441.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Zbigniew L. Pankowicz (IRDT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linguistic Theory
Computational Linguistics
German-English Machine Translation
METAL System
FORTRAN/LISP Implementation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Report constitutes a linguistic documentation of the third generation machine translation system designated METAL (Mechanical Translation and Analysis of Languages, or METALanguage). Section I of the Report presents an overview of the system (underlying philosophy, general description of system's components). It also includes a description of 14 programs (translation phases), used in the original FORTRAN implementation, and a description of the system's LISP version (8 translation phases). The remaining sections describe in detail each phase in the German-English machine translation process. The description

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

for a phase gives examples of rules used at this stage, and annotates the features and possible feature values used by the program, as well as the operations that may be performed at that point. Section II describes the German lexicon and the feature system used for German. Section III explains how grammar rules are applied to the lexical entries to find a syntactic parse of the sentence. Section IV states how the surface structure parse tree is converted into a deep structure. Section V deals with lexical collocations with regard to the recognition of German idioms. Section VI dercribes the normal form grammars, used to change the German deep structure tree into an intermediate structure for translation, and then the mapping of this tree into an English deep structure. The contents of the English deep structure tree are delineated in Section VII. The feature system used by the system for English lexicon material is specified in Section VIII. Syntactic rules for English are described in Section IX, and Section X explains the production of English output from the English syntactic tree.

Preface

This report describes the machine translation system of the Center in its current state. Sections of the report indicate clearly the dynamic features of the system. It was designed in this way. From the first, linguistic descriptions and computer programs were developed independently of each other. As advances were made in software systems, these could then be introduced. The most recent modifications are illustrated in this report by the accounts of LISP in contrast with FORTRAN implementation. Further modifications and improvements can be expected. Because of the system design, however, this will not require disruption of its operation, whether the changes concern linguistic or programming matters.

The system represents the cumulative efforts of the many scholars who have been on the Center staff, and also contributions of others in linguistics and computational study. Machine translation has always been viewed in the Center as a technological activity which would be improved by advances in computer hardware and software, and in our understanding of language.

As the system is improving, spin-offs become more notable. The programs can be used in other linguistic research, for example. And members of the staff have the opportunity of upgrading their competence in the computational and linguistic fields.

As a component of a university, the Center has always held such education as one of its major aims, education which involves contact among specialists in the humanities, sciences and engineering. In addition to noting such interdisciplinary work, we would like to acknowledge the contributions of all who have participated in the development of the system. Their achievement of a translation system then illustrates cooperation across disciplines and departments which several decades ago had little contact with each other.

Winfred P. Lehmann, Director

TABLE OF CONTENTS

EVALUATION

The Report describes in detail the linguistic performance of the third generation machine translation system designated METAL (Mechanical Translation and Analysis of Languages, or METALanguage). The design of the system is tripartite, i.e., consisting of three components, a lexicon, a grammar and a processing algorithm that can be used with the linguistic description of any given language. By comparison, the design of a second generation machine translation system is bipartite, i.e., consisting of two components, a lexicon and a translating algorithm. The rules governing the translation process in such systems are directly incorporated in the algorithm, thus restricting the system's performance to a unidirectional translation involving two specific languages. Consequently, any changes that have to be made in the linguistic data base require corresponding changes in the system's programs. Past experience with bipartite systems also shows that a high quality translation cannot be achieved by directly translating a source language surface text into a target language surface text. Optimization attempts by means of linguistic and logical mechanisms embodied in bipartite systems are counterproductive and self-defeating, because systems so designed are not inherently improvable.

Translation performance of the METAL System is based on lexical, syntactic, semantic and pragmatic (paralinguistic) types of information, which corresponds to performance criteria imposed by tradition upon a competent human translation. Linguistic problems that constitute indispensable prerequisites to a high quality translation are accounted for in the System in a methodical manner reflecting a continuous interaction among information types indicated above.

Assumption of a universal semantic base, or interlingua, is the cornerstone of the METAL System. This assumtion states, in effect, that the intrinsic meaning of a linguistic utterance is the same regardless of its expression in any particular language. The translation process is consequently divided into analysis of source language, transfer (source language to interlingua, interlingua to target language) and synthesis in target language. This translation concept is advantageous because the linguistic description of any source language in the System does not depend on considerations of its translatability into any particular target language. When the linguistic description of an additional language is incorporated in the System, translation can be performed from that language into any other language

already accounted for in the System. Furthermore, the translation process is not restricted to one direction, but can be performed bidirectionally, i.e., from source to target language and vice versa.

Algorithms of the METAL System are generalized linguistic data processors and have no concept whatsoever of languages they operate with. Programming strategy is, therefore, completely independent of specific characteristics of any particular rules; they only expect the rules to be expressed in the required format. Programs are written in higher order languages, i.e., FORTRAN and LISP, which imparts a high degree of computer independence to the System and guarantees the software immunity in transition from one computer to another. The advantage of this programming philosophy consists in the fact that the adaptation of the System to machine translation of different language pairs does not entail any changes in programs.

Dictionaries of the METAL System are based on a comprehensive lexicographic classification scheme; it permits a quite exhaustive classification of words in terms of their morphological, syntactic, semantic and pragmatic properties, including properties of the environments in which they occur. This information is exploited by the System's components to disambiguate a source language word in its context and to select the appropriate target language equivalent.

Grammars of the METAL System consist of terminal and non-terminal rules. The former describe internal features of the word and its external functioning characteristics. The latter describe the patterns occurring in sentences and parts of sentences. The format of the rules is not restrictive; any kind and amount of information can be stated within its framework. The format is furthermore convenient and easy to learn; it permits the encoder to write rules in a way he is accustomed to from his school training in grammar. Since this effort does not require changes in programs or special skills, updating of the existing linguistic descriptions within the System can be performed as an in-house effort without external assistance under contract.

The METAL System exemplifies the evolutionary progress in the state-of-the-art beyond its current limitations, and the Report documents an important milestone in machine translation R&D. The System is based on cumulative advances achieved during the past decade in conceptualization of machine translation, linguistic theory, computational linguistics and ADP hardware/software technology. The System is open ended and, therefore, inherently capable of capitalizing on any other such advances in the future.

ZBIGNIEW L. PANKOWICZ
Project Engineer

I. THE LRC TRANSLATION SYSTEM

The LRC MT system is a configuration of algorithms and grammars designed to derive from a given sentence the meaning(s) of that sentence; and, given a particular meaning, to derive all sentences with that meaning. Like other grammatical models, it assumes three levels of representations of a sentence: the surface description, the deep description, and the semantic description. The latter two are referred to here as "standard readings" and "normal form readings", respectively. The following graph corresponds to these levels of representations.

Semantic
Description
↕
Deep
Description
↕
Surface
Description
↕
Text

Note that the arrows are bi-directional; that is, it is a system of effective procedures for the derivation of the meaning of a text from that text, and the derivation of a text from the meaning of the text.

The LRC system assumes that the meaning of a sentence is independent of the language in which the sentence is uttered; that is, it assumes a universal semantic base. Every arrow in the above graph corresponds to a pair of grammars and algorithms which process these grammars. The up-arrow corresponds to the input language pair; the down-arrow, to the output language pair. If the grammars used for input and output language are identical, we call the process paraphrasing; if the grammars are distinct, we call the process translation.

Linguists have argued convincingly in the past that mechanical translation from surface structure directly to surface structure cannot produce satisfactory translation, and that first a deeper representation must be derived which can then be used to translate into a corresponding deep representation of the output language. To validate this argument it may be sufficient to point out that often translation requires a change in part of speech of the translation equivalents. For example, German--

"er trinkt gern"

where the German adverb "gern" becomes the English verb "like"

"he likes to drink"

or--

"die Bundesrepublik Deutschland"

"the Federal Republic of Germany"

where the German noun "Bund" is translated as the English adjective "Federal".

Similarly, the occurrence of lexical collocations requires translation performed on a level deeper than the surface description. These idioms and idiom-like locutions are dictionary entries which consist of more than one word and whose syntactic interpretation and meaning cannot be derived from that of the individual words of the collocation. E.g.: "in Frage stellen" = "to question"; or in English, "make up one's mind" = "decide". The frequency of their occurrence is higher than generally assumed. A study done previously by LRC found 135 distinct lexical collocations in a 1500-sentence scientific corpus.

A. General Description of Grammars

Whenever we use the expression 'grammatical description' with respect to MT, we are really talking about two grammatical descriptions simultaneously. The first description is the one provided by the linguist. These descriptions are stated in very general terms, for example, for German:

> a noun phrase may consist of a determiner followed by a noun; the two constituents must agree in gender, number and case. The noun phrase acquires this common gender, number and case; moreover, the noun phrase has all the properties of the noun.

The grammar formalism permits the statement of such general syntactic and semantic descriptions by means of the set-theoretical operations intersection, union, and complement. These are used to express relations which may hold between the values of specified subscripts.

The grammars written by the linguists are context-free grammars with complex symbols. The corresponding processing algorithm operates with the grammatical description given by the linguist and generates the actual grammatical description of a constituent. The description provided by the algorithm is far more precise because it mentions all the properties of the constituents which are not explicitly mentioned in the rule.

Since one grammar rule thus represents an abbreviation of a large but finite number of descriptions provided by the algorithm, the advantages of using such grammars with context-free analysis are not overcome by complex symbol grammars. Thus the constituents to be interpreted by a rule have to occur in a text in the order specified by the rule. This results in a proliferation of their constituents.

With two exceptions the LRC algorithms are direct substitution analyzers. They perform grammatical analysis from bottom-to-top, processing a text from left-to-right. At each text position T they apply all rules which provide an analysis for every text span beginning to the left of T and including T. Such analyzers have been described in detail at various places. In this description of the LRC algorithms we will mainly point out the interrelationship between grammatical description and algorithmic performance.

1. The Anatomy of Rules

The LRC German-English translation system utilizes a total of twelve grammars and/or dictionaries. Of these twelve components, seven apply to the input language, German, and five apply to the output language, English. Each of the grammars can be classified as one of three types, Dictionary, Syntax, or Transfer. The grammars used in the translation system are:

German Dictionary Grammar [GFG-D, GVD]
German Word Grammar [GFG-W]
German Syntactic Grammar [GFG-S]
German Macro Choice Grammar [GFC]
German Lexical Collocation Grammar [GSG-W]
German Standard Grammar [GSG-S]
German Transfer Grammar [GNG]
English Transfer Grammar [ENG]
English Dictionary Grammar [EFG-D]
English Word Grammar [EFG-W]
English Syntactic Grammar [EFG-S]
English Macro Choice Grammar [EFC]

The rules in all of the grammars are essentially "re-write" rules in which the term or combination of terms occurring to the right are rewritten as the leftmost term (immediately following the rule number), e.g.,

```
C 30105   V N          V A        V N
          ^ 3          P          B
```

Exclusive of the identifying numbers, the format has a basic four-way division. Horizontally, it is composed of the left-side term and one or more right-side terms. Vertically, it has a top line in which the constituents are named, under which any features pertinent to a constituent are listed.

```
          Left      Right
          Side      Side
Rule No.  Term      Term(s) . . .
________  _______   ______________
C 30105   V N        V A      V N  |< Constituents
          ^ 3        P        B    |< operators; modifiers;
                                   | subscripts and values
```

This rule contains three variable constituents, symbolized by "V". Here the initial term of rule C 30105 is "N" (Noun-stem), the second term is "A" (Adjective-stem), and the third is another "N". The rule states: the combination of an adjective stem plus a noun stem is to be analyzed as a noun stem. It is the equivalent of a phrase structure rule N -> A + N. (On the second line the "P" indicates that the adjective must be the initial constituent in the word, and the "B" signifies that for this rule to apply there cannot be a blank or punctuation between the adjective and noun stems. The "^ 3" signifies that the resulting compound is to carry the same syntactic and semantic features as the third term, the original uncompounded noun stem. noun stem.)

Most of the grammar rules are more complex than the above in that they contain additional information and conditions. One of the more noticeable aspects of most rules is the presence of coded "features" conveying syntactic and semantic information.

```
C 21652  V N               * PHYSIK
         + CL(1)
         + GD(F)
         + TY(AB)
         + NU(S)
```

The subscripts and values in rule C 21652, above, convey the information that the German noun for "physics" is of morphological class 1, CL(1); its gender is feminine, GD(F); its semantic type, abstract, TY(AB); and its number, singular, NU(S). The other major type of feature, the operators, will be discussed in the following section.

Features may be used to indicate a wide range of linguistic and paralinguistic information such as type of subject or number of objects possible, kind of discourse, area of provenience, style, linguistic frequency, etc., in order to limit the application of particular rules in parsing and generation of sentences. The system's algorithm permits the incorporation of additional features of this sort into the rules whenever the necessary linguistic information is accumulated and additional distinguishing attributes are required by the particular texts analyzed.

Three types of constituents, corresponding to nodes in a sentence phrase-marker, are used in the rules. Terminal constituents, which represent the actual surface structure of the sentence, are marked by the symbol "*", as in rule C 21652 above. Non-terminal constituents which are created from lower level constituents are referred to as "variables" (symbolized by "V") since each may represent any number of surface lexical items, as in rule C 30105 above. There are also "dummy" constituents, symbolized by "D", which are inserted by the rules and do not correspond to particular surface constituents. I.e., dummies may be introduced in order to carry tense and voice information or to indicate boundaries between constituents.

B. General Description of the System

1. FORTRAN Implementation

The prototype of the translation system (as well as many of the supporting programs) was initially implemented using FORTRAN for a CDC 6600. It employed a total of fourteen programs. Six of the programs could be classified as lexical scanners or parsers and are referred to as Analysis programs. Another six programs make use of various well-formedness conditions to eliminate irrelevant or inapplicable information, and are referred to as Choice programs. Of the remaining two programs, File Entry Construction is a translator and Lexical Spellout is an output formatting program.

Programs used in Translation

- Dictionary Analysis
- Dictionary Choice
- Word Analysis
- Word Choice
- Syntactic Analysis
- Syntactic Choice
- Standard Analysis
- Standard Choice
- Transfer Analysis
- Transfer Choice
- File Entry Construction
- English Syntactic Analysis
- English Syntactic Choice
- Lexical Spellout

a. Surface Component

Surface analysis accepts a text of the input language and processes it with the grammars for that language to produce as output a set of standard sub-trees which are representative of the deep structure of the language.

Surface analysis is executed in three phases, the dictionary phase, the word phase, and the syntactic phase. Each of these phases consists of an analysis stage and a choice stage. The surface component thus consists of six sub-components: Dictionary Analysis, Dictionary Choice, Word Analysis, Word Choice, Syntactic Analysis, and Syntactic Choice.

1) German Dictionary Structure

Dictionary Analysis seeks all sequences of contiguous stems and endings which cover a word, beginning with the first letter. The reason for recognizing all stems and endings is to account for the ability of a number of languages to form compounds. This particular property prevents the application of the longest-span method, a procedure which would lead to different results for German, depending on whether dictionary analysis was performed from left-to-right or right-to-left.

Dictionary Analysis is performed with the dictionary grammar which is stored as a set of 64 trees, the roots of which represent the initial symbol a dictionary entry can begin with. Each root is connected through n branches with daughter nodes where each such daughter node represents a letter which occurs as the second letter after the 'root' letter in a dictionary entry. An additional daughter node is added after each node which ends a word.

Dictionary Choice determines which stem(s) and/or endings cover a word continuously, beginning with the first letter of a word and ending with the last letter of a word. Entries which are not part of such a contiguous complete sequence are deleted.

Besides computing the dictionary entries to be retained for subsequent analysis, Dictionary Choice can be influenced by a preference operator, given in certain dictionary entries. The effect of a preference operator is that all well-formed subsequences within a dictionary entry are deleted. This preference operator was mainly attached to rules for German endings in order to prevent the occurrence of intermediate, non-productive interpretations. Spans which are not covered by a complete sequence of interpretations are assigned a category symbol K, signifying 'unknown word(s)'.

2) German Word Structure

The word component is designed to--

- find all segments or sequences of segments which are well-formed according to the word grammar of a language, and record that description;
- delete all segments or sequences of segments which are not well-formed;
- determine those text positions where a sentence or clause boundary could theoretically occur;
- identify the nodes on which Syntactic Analysis is to continue building.

Word Analysis applies all rules applicable to every text span available at each position in the text. It operates with the word grammar of a language, whose initial symbol, for all languages, is WORD. Any text span covered by an analysis dominated by WORD is for that reason well-formed according to the word grammar of that language.

Word Choice performs the four functions listed above by--
- retaining all readings dominated by WORD;
- destroying all interpretations not dominated by WORD, including the artificial symbol WORD;
- inserting potential sentence boundaries into the text spans according to the specifications of the Word Analysis rules applying to that text span; and
- flagging every node which was not immediately dominated by by WORD.

3) German Syntactic Structure

Syntactic Analysis seeks to find all readings of a text span which are well-formed according to the syntactic grammar of a language, i.e., readings dominated by the grammar's initial symbol S. The Syntactic Analysis algorithm, now identical to the Word Analysis algorithm, operates with the syntactic grammar of a language.

Using information contained in the Syntactic Analysis rules (subsequently called the "Choice Statement") and in the separate Syntactic Choice Macro Grammar, Syntactic Choice serves to disambiguate words on the basis of information contained in a sentence, and to determine the semantic function of each constituent and its semantic relations towards other constituents. It discovers discontinuous constructions and makes them contiguous, introduces new dictionary terms which did not occur in the text, and deletes words and non-terminal constituents which can be predicted or which become features of their verb.

b. Standard Component

The standard component derives the standard reading of a sentence (or readings, in case of ambiguity) which can subsequently be interpreted by the meaning assignment grammar or normal form grammar. The input for the standard components are the set of tentative standard trees provided by the Syntactic Choice phase.

Its output is a set of well-formed standard trees, that is, trees which are dominated by the symbol S. If a sentence was structurally ambiguous and no selection was made during Syntactic Choice, more than one standard reading will be generated. (Lexical ambiguity does not result in multiple standard analyses.) Standard Analysis is performed in three phases: Lexical Collocation, Standard Analysis, and Standard Choice.

1) Lexical Collocation

Lexical Collocation analyzes for sequences of terminal symbols which are idioms or quasi-idiomatic, without affecting the structural interpretations of the individual component since these might be needed to determine possible transformations. In a surface text, elements of a lexical collocation can occur in any order and at any distance, that is, they may occur discontinuously. After Syntactic Choice, however, we can determine precisely, if a lexical collocation occurs, which order the individual elements will occur in, their syntactic superstructure, and which constituents, if any, occur between them, obligatorily and optionally.

2) German Standard (Deep) Structure

Standard Analysis is designed to construct a complete syntactic tree dominated by S which represents the deep or underlying structure of the sentence. The Standard Analysis algorithm is identical to the Lexical Collocation algorithm, but it only operates on constituents which were unflagged by Syntactic Choice. Any constituents not unflagged could not be built upon. Standard Analysis, which operates with the standard grammar of a language, is very small in size. It is because of the predefined order in which the constituents occur, that the various syntactic and/or semantic relations that must hold between constituents can be expressed economically by means of a small number of grammar rules. The purpose of standard grammar is basically twofold:

- it assigns the correct structure to sentences which contain lexical collocations, and
- it regenerates the syntactic superstructure that was destroyed during Syntactic Choice.

Standard Choice deletes all readings which are not well-formed according to the standard grammar, i.e., not dominated by the symbol S. It is similar in operation to the Syntactic Choice phase. Standard Choice does not introduce additional dummies

nor does it delete rule terms. It does perform additional disambiguation and permits the rejection of whole trees if they compete with trees which dominate a lexical collocation with the feature LX(P) for 'lexical collocation with preference'. The Standard Choice phase does not operate with a separate standard choice grammar. All instructions executed by Standard Choice are part of the standard rules. Choice statements which are attached to structures which were already constructed during Syntactic Analysis and consequently executed by Syntactic Choice are not repeated by Standard Choice.

c. Transfer (Translation) Component

Transfer Analysis associates with each word in a standard reading its word sense -- if the word was disambiguated -- or its word senses if the word is still ambiguous. Similarly, it associates with each sub-tree or with connected sub-trees (mother, daughters, and possibly daughters and descendants of daughters) a functional interpretation. These interpretations are referred to as normal form expressions. The Transfer Analysis phase operates with the transfer grammar of a language and determines for each standard text the normal form readings applicable to that text.

Transfer Choice eliminates all normal form readings which do not take part in a contiguous coverage of the standard tree structure. Again, a special P-operator attached to a transfer rule permits Transfer Choice to select such readings in preference to other transfer readings covering the same structure.

d. Standard Production Component

Standard Production associates with the normal form reading of a sentence the standard trees of all English output sentences having the same normal form reading. In other words, the object is to find all English sentences which have the same meaning as the German sentence. Secondly, Standard Production tries to arrange the constituents of such sentences in an order which closely reflects an acceptable English surface order. The Standard Production Component consists of two sub-components: File Entry Construction and Output Syntactic Choice.

1) File Entry Construction

File Entry Construction associates the normal form readings provided by the analysis of a German input sentence with all well-formed English standard trees (dominated by the symbol S) with the same meaning as the input sentence. At the same time, File Entry Construction tries to produce translation sentences whose structural description is similar to the description of the German sentence, in order to reduce the number of translations provided for each input sentence. It is executed in two phases, Interlingual Mapping, and Synthesis (English Syntactic Analysis).

Interlingual Mapping associates with the German normal form expression all English sub-trees interpreted by the corresponding English normal form expressions. At the same time, it checks whether the retrieved standard sub-trees can be connected.

The Synthesis portion of File Entry Construction checks that the superficial standard trees are indeed well-formed according to the standard grammar of the language. Those sub-trees which do not satisfy this condition are deleted. At the same time, Synthesis checks that the nodes of the remaining trees have the subscripts and values which were stated in the English transfer rules. The output of Synthesis is a well-formed English standard tree (or set of trees in cases of structural ambiguity of the input text).

2) English Surface Structure

Output Syntactic Choice selects the values of lexical dummies that need to be printed by Lexical Spellout, and it determines the order in which the constituents are to be printed out. It utilizes the English Standard Choice grammar as well as the choice instructions contained in the English standard rules.

e. Sentence Production Component

The Lexical Spellout phase is simply a text-formatting program which accepts the output of Syntactic Choice and produces English written text from it.

2. LISP Implementation

Currently, however, research is being directed toward a more efficient implementation using LISP, a newer programming language better suited for string manipulation. Using the same dictionaries and grammars, the LISP implementation requires only eight levels in translation, since many of the operations originally done in Choice can now be accomplished during the analysis phases.

a. Morphological Analysis

The first phase of the translation process locates the most likely morphemic constituents of an input sentence. The input sentence is a string of words and punctuation. The other input to this phase is a morpheme tree containing a lexical rule number, a word class, and usage data for each morpheme. The usage field contains information as to where the morpheme may occur in a word (e.g., as a suffix). The tree is organized so that when traversed with a word, the longest morpheme from the start of the word is found first and then the remainder of the word is analyzed in the same manner. If the remainder can not be analyzed, the analyzer backs up to the next longest morpheme and tries again. This phase succeeds if every word in the sentence can be morphologically analyzed, with each punctuation mark remaining unchanged. Independent of the morphological analyzer a set of functions exists that builds the morpheme tree from a list of morphemes or augments the tree with additional morphemes.

b. Word Analysis

The Word Analysis phase uses the output from Morphological Analysis and applies grammar rules to the morphemes, building them up to structures that are the equivalent of case predicates. This phase first transforms the input so that it may be used as input to a parser. The parser uses the Cocke-Younger-Kasami algorithm to build the parse tree from the input and grammar rules. The grammar rules contain: a left-side, the result; a right-side, from which the result is built; a set of tests that must be true for the rule to apply; and a set of operations used to build the new left-side node.

c. Syntactic Analysis

Syntactic Analysis applies grammar rules to the "case predicates" from Word Analysis and builds up to the sentence symbol. Syntactic Analysis first builds clauses from the case predi-

cates, and then a sentence from the clauses and punctuation. The parser uses the Cocke-Younger-Kasami algorithm again. Macros are applied during this phase to perform tests and assignments that are common to many clause rules. A transformation orders the lower level sons of the case predicates, including dummy nodes, and unflags these nodes so the next phase may operate on them. This phase succeeds when the first output from Word Analysis is built to the sentence symbol.

d. Standard Analysis

Standard Analysis applies grammar rules using the Cocke-Younger-Kasami parser to the unflagged nodes which were ordered as the result of the transformations applied by Syntactic Analysis. The purpose of Standard Analysis is to apply these rules and build up the transformed structure back to the start symbol. This phase returns the new root node.

e. Two Transfer Phases

The purpose of the Transfer phases is to take the standard tree of the source language and change that tree into a standard tree for the target language. The first transfer phase applies a grammar rule to each node of the input tree to transform the tree into a deep structure representation. The second transfer phase applies a grammar rule to each node in the deep structure tree to transform it into a standard tree for the target language. This new tree contains the rule numbers for the grammar rules that a non-terminal node should apply to its sons and the lexical rule number for terminal nodes. The root of the new tree is the result.

f. File Entry Construction

File Entry Construction applies the grammar rules specified in the tree to the sons of non-terminal nodes and instantiates the lexical entries for the terminal nodes. The clause entries will order the case predicates for target language output. The root of the tree constructed is returned.

g. Lexical Spellout

Lexical Spellout uses the tree from File Entry Construction to produce target language output. This phase prints the terminal nodes of the tree from left to right, concatenating the morphemes of a single word together.

II. GERMAN LEXICON

A. German Dictionary Grammar

The purpose of Dictionary Analysis is to find all sequences of contiguous stems and endings which cover a word, beginning with the first letter.

The reason for recognizing all stems and endings is to account for the ability of a number of languages to form compounds. German is well known, if not infamous, for its noun compound formation which theoretically permits the generation of an infinite number of nouns. This particular property prevents the application of the longest-span method used for languages like English, French, and Russian at such institutes as TAUM (Montreal) and GETA (Grenoble). Such a procedure would lead to different results for German, depending on whether dictionary analysis was performed from left-to-right or right-to-left. The word "VORKRIEGSPIONIEREN" would thus either be segmented into "VORKRIEGS" + "PIONIEREN" (left-right), or "VORKRIEG" + "SPIONIEREN (right-left).

Not all stems or endings which occur in a word are recognized by the algorithm. Thus the sequence "ESSER" or the letter "E" (after "GRO") are not recognized as a noun or an ending respectively, since no stem or ending terminated at the letter "O" in "GROESSER". The recognition of such intermediate spans as dictionary entries is prevented by means of preconditions and post-conditions which can be associated with each dictionary entry. Preconditions are of four kinds:

a- the predecessor must have been a marginal symbol (blank and/or punctuation mark)
b- the predecessor must have 'set a morpheme boundary'
c- the predecessor must have been of type a or type b
d- the predecessor is ignored.

Post-conditions associated with the dictionary entry have to be satisfied by the preconditions of the rules applying to the string following such an entry. The restrictions which can be set are:

e- set a marginal symbol boundary
f- set a morpheme boundary (stem or ending boundary)
g- do not set a morpheme boundary.

That the letter G does not set a morpheme boundary prevents "ROESSER" ("horses") from being recognized, since "ROESSER" requires a precondition of type b. The same condition also

prevents "ESSER" from being recognized. The letter "E" (after "O") has a precondition of type b; the pronoun "ER" requires a precondition of type a.

Dictionary Analysis is performed with the dictionary grammar which is stored as a set of 64 trees, the roots of which represent the initial symbol a dictionary entry can begin with. Thus all words beginning with A occur in the tree with root A, those beginning with B occur in the tree with root B, etc. Each root is connected through n branches with daughter nodes where each such daughter node represents a letter which occurs as the second letter after the 'root' letter in a dictionary entry. An additional daughter node is added after each node which ends a word.

The analyzer follows and discovers all possible interpretations 'simultaneously'. Thus, when the analyzer discovered the adjective stem "GROESS" in the example above, it continued the analysis which eventually discovered the noun "GROESSE" but at the same time started the analysis which discovered the endings "E" and "ER".

Dictionary Analysis uses only the information found in the right side of dictionary rules, namely those found in the German Surface Dictionary - Word Grammar [GFG-D] and the German Verb Dictionary [GVD].

In the FORTRAN version, Dictionary Choice processes the output of Dictionary Analysis. The purpose of Dictionary Choice is to discover all stems and/or endings which cover a word continuously, beginning with the first letter of a word and ending with its last letter. It uses no additional linguistic data, i.e., no special grammars or dictionaries are employed at this level, but it checks whether the conditions of any Choice operator are met. Entries which are not part of such a contiguous complete sequence are deleted.

Besides computing the dictionary entries to be retained for subsequent analysis, Dictionary Choice can be influenced by a preference operator, given in certain dictionary entries. The effect of a preference operator is that all well-formed sub-sequences within a dictionary entry are deleted. Assume that the dictionary contains "RUECKSTOSS", "STOSS", "KRAFT" and "STOSSKRAFT" where "STOSSKRAFT" has a P operator. If the text had "STOSSKRAFT", the entries "STOSS" and "KRAFT" would be deleted after Dictionary Choice. If "RUECKSTOSSKRAFT" occurs in a text, however, the entries for only "RUECKSTOSS" and "KRAFT" remain after Dictionary Choice.

This preference operator was attached to rules for German endings mainly to prevent the occurrence of intermediate, non-productive interpretations. Thus the preference operator attached to the ending "EN" deletes the endings "E" and "N" in forms like "GROESSEREN", preventing an interpretation of "GROESSERE" during Word Analysis.

Spans which are not covered by a complete sequence of interpretations are assigned a K-reading, described later, which stands for 'unknown word(s)'.

Following the basic rule format, dictionary rules consist of a rule number, a left side term, and a single right side term. The top line of the left-side term in a rule is a "variable" indicating the kind of morpheme. (E.g., N = noun stem; NU = numerical classifiers, including number words, digits, and words like dozen; SCH = the affix "sch"; EE = the inflectional ending "e"; FIN = finite verb.) The right-side term represents the actual surface string analyzed by the rule, as for example "vertrag" in the following:

```
C 22546        V N                          * VERTRAG
               + CL(4)
               + GD(M)
               + TY(AB)
               + FC(LA,MIT)
               + TC(LA,AB,HU+CO)
               + IO(LA,O)
               T 1.4
```

Here the "*" indicates that "vertrag" is a surface constituent. The rule specifies that this string is analyzed as a noun variable, "V N", and the group of syntactic and semantic features is assigned to the node which is created. (The T or trace operator at the bottom of the list of features will be discussed later.)

The information utilized at this level is the string of characters representing the morpheme in question and any "operators" present. On the right side these operators, usually printed directly below the string, are the B, the E, the P, the absence of an operator symbol, and the F.

B : indicates that the string must not be immediately preceded by a blank or punctuation mark, i.e., the string must be preceded by another string (to which it is suffixed). This operator is generally used with derivational and inflectional endings, as in:

```
C 1     V EST          * EST
D 35                   B
```

E : the segment analyzed by the rule may apply anywhere, regardless of the nature of the preceding character; in short, there are no contextual restrictions. Periods, for instance, can occur not only after words but between digits in decimals, between letters in library call-numbers, and even between other periods in marking ellipsis.

```
C 43    V PNCT         * .
        + TY(PRD)      E
```

P : the rule may only apply to a string of text which is preceded by a blank or punctuation mark; the morpheme in question cannot take another morpheme as prefix.

```
C 9537     V PRFX       * NACH
           + PX(NACH)   P
```

When no operator is specified below the right-side string, the morpheme must be immediately preceded by another interpreted string; the rule could not apply unless the immediately preceding character completes a string.

```
C 21962    V N          * SCHALL
           + CL(17)
           + GD(M)
           + TY(NT)
```

F : the rule may only apply to a string of text which is followed by a blank or punctuation mark; the morpheme in question cannot take another morpheme as suffix or occur as the initial element in a compound, e.g., pronouns or certain final suffixes.

```
C 40047    V CONJ       * WEIL
           + CJ(S)      P
           + KT(S)      F
```

Like the F-operator on the right side of a rule, a \ on the left also places a constraint on the characters which may follow a morpheme. A rule with a \-operator may only apply to a string which is followed by a marginal element, represented by the M-symbol (see below).

```
C 6072      V AV          * ETWAS
              \             P
```

On the left side of a rule, usually near the bottom of the list of subscripts and operators, an M can occur. It represents blanks or punctuation, and marks potential syntactic junctures. Dictionary Choice discards all file entries which are not flanked by file entries containing M in their left-side term.

```
C 53     V BLANK          *
         M                E
```

In the event that a sentence contains a span of text bounded by M-symbols which is not analyzable by Dictionary Analysis, one has the option of generating a special rule for such a string not included in the lexicon. This option, the K-option, is one of the available input parameters for Dictionary Choice. If the option is "on", the unprocessed string will be assigned the left side node "K" and will be included in the analysis for translation. This option is generally used to account for proper names or quotes from a foreign language which would not usually occur in the lexicon. They will be translated with the same surface value as in the original text.

```
C274     V K              * NEWTON
```

There is also a P-operator which can appear on the left side of a rule. It is an option which allows the system to select (or give preference to) a particular analysis of a span by discarding any other sequences or file entries covering the same span as the rule with this P-operator. Generally the option is used for sequences of characters in the input text which represent a potential source of ambiguity as a result of being identical with series of shorter sequences also found in the lexicon. The preference operator is especially important in entries such as

```
C 40033        V CONJ                  * DASS
               + CJ(S)                 P
               + KT(S)                 F
               + WD(TH)
               + TYP (CMPL,CONS,FINAL)
               T 1.4
               P
```

Since the dictionary contains rules for "das" and for the suffix "s", it is possible that two analyses may have been created. The preference operator guarantees that only the analysis based on the rule illustrated above is retained by Dictionary Choice when the string DASS is both preceded and followed by a blank or punctuation mark.

1. Conflation of Dictionary Rules

Individual German and English verbs are initially entered into the grammar as many times as a form has different meanings and/or selection restrictions. However, it has been found useful to conflate the multiple entries for a single morpheme, thus eliminating redundant information and reducing the size of the storage necessary and the number of internal analyses. The special dictionary rule conflation program takes a set of dictionary rules-- R1, R2, ... Rm, each with the same rule number, left-side category symbol, and right-side, where each Ri has a set of left-side subscripts with N columns-- and constructs one rule having T columns where,

T = Sum Ni, for i=1,m

A list of subscripts is input to the program and only the subscripts appearing in this list are columnized (each column of values being separated from the other by an apostrophe). If no columnized subscript occurs in one of the Ri rules, the value LA (null) is inserted at that position. If a subscript is not to be columnized, it is conflated to a single entry and each of the Ri subscripts, if they differ, are separated by commas.

Any "trace" information contained in individual rules is carried over to the conflated rule. (The T, or trace operator, is used at later levels in deleting columns of values which are not appropriate for the particular sentence usage.)

For example, note the two rules applicable to "beweg":

```
C  4082          V  V             *  BEWEG
D  1             +  CL(56)
                 +  PX(FORT)/
                 +  FS(N)/
                 +  TS(AN)/
                 +  FO(A)/
                 +  TO(R)
                 T  1.2

C  4082          V  V             *  BEWEG
D  2             +  CL(56)
                 +  PX(VORW)/
                 +  FS(N)/
                 +  TS(PO)/
                 +  FO(A)/
                 +  TO(R)/
                 +  OA(DIR)
                 T  1.2
                 T  1.6
```

The subscripts to be columnized are PX, FS, TS, FO, TO, and OA. The resulting conflated rule is:

```
C  4082          V  V             *  BEWEG
                 +  CL(56)
                 +  PX(FORT'VORW)/
                 +  FS(N'N)/
                 +  TS(AN'PO)/
                 +  FO(A'A)/
                 +  TO(R'R)/
                 +  OA(LA'DIR)
                 T  1.2
                 T  1.6
```

The conflation eliminated the redundant statements of information. Though this is a less readily comprehensible form of the rules for the human user, it poses no problem for the computer. In fact, the lessening of redundant information it must make its way through actually speeds up its operations.

B. The Semo-Syntactic Feature System

Much of the "power" of the LRC translation system lies in the sets of descriptive semo-syntactic features which are associated with each lexical entry. These features are used to characterize the syntactic and semantic environment in which a lexical entry may occur. The features differ from one part of speech to another, and so will be discussed separately for each morphological category.

1. German Noun Features (Category Symbol = N)

Because German nouns carry inflectional markers for case, number, and gender, their analysis can be of significant value in determining the relationship among the constituents in the sentence. In addition to the obvious syntactic co-occurrence restrictions between nouns and other elements of a sentence, investigation has shown that accurate translation requires the testing of certain semantic relationships. A verb such as "sprechen", for instance, would normally require a human subject. Similar restrictions apply between nouns and their adjective modifiers.

Hence to ensure quality translation it is necessary to encode a considerable number of syntactic and semantic features for nouns. Such features serve to disambiguate sentences and to prevent unlikely parses while establishing the syntactic relationships among sentence elements. The basic features so far found to be relevant are:

CL = class
GD = gender
SX = sex
TY = semantic type
FM = form of noun
DF = derived form
FC = form of complement
TC = type of complement
TO = type of object
OB = object of nominal
IO = interpretation of object
CAN = canonization
TT = tantum noun
RA = required adverb
OA = optional adverb
LC = lexical collocation
LG = source language for loan word
FR = future research
CP = capitalization

(All nouns carry features for class, gender, and type. The other features may or may not be present, depending upon the particular noun and its derivation.)

CL (class) specifies the inflectional class of a noun stem, class agreement being required between nouns and inflectional endings. The values for class are numerical, i.e., the classes are designated by numbers. Thus far, sixty-four paradigmatic classes have been identified for nouns in German. They are listed in the chart at the end of this sub-section.

GD (gender) represents grammatical gender. This is a normal morphological feature of German and is distinct from natural gender, which is covered under SX (sex) where relevant. Grammatical gender is important in identifying relationships between nouns and potential noun modifiers in German. It often provides an important clue in determining syntactic structure since nouns and their modifiers must agree with respect to gender.

M = masculine
F = feminine
N = neuter

SX (sex) is used for natural gender, as opposed to grammatical gender. Usually the feature is not expressed for those nouns which lack inherent sexuality. In German nouns it is only used when the sex (and hence pronominal agreement) may differ from the grammatical gender. For example, German "Maedchen" is a neuter noun with respect to grammatical gender yet represents a female individual. The values of SX are:

FE = female (DAS Maedchen)
MA = male (DAS Maennchen)
MA or FE = both (DAS Dickerchen)

TY (semantic type) represents the semantic class of the noun. The distinctions included under this subscript are those values which are necessary in certain well-formedness conditions between nouns and verbs and between nouns and certain adjectives. Such distinctions are frequently useful in disambiguation. For example, a noun such as "page" is ambiguous in English. It may be either a human being or an inanimate. If it is used with a verb such as "speak", one can readily disambiguate the nominal since the verb "speak" requires an animate subject.

The values which have been used thus far include:

E = entia (anything)
PO = physical object
AN = animate
HU = human
AL = animal
PL = plant
IN = inanimate
BP = body part
MA = machine
NT = non-tangible (e.g., electricity, light)
AB = abstract
CN = count noun
UN = unit
QU = quantity
MS = mass
NM = name
CO = collective
TM = time
T = title
F = female
M = male

These values are nested in a tree structure for purposes of checking agreement during parsing. A particular value is subsumed under any higher value from which it branches. Thus, for example, a rule requiring an animate nominal would apply to TY with the value HU or AL.

```
                                 E
                                 |
                                 |
               ----------------------------------------
               |            |            |            |
               |            |            |            |
               NT           PO           QU           AB
                            |                         |
                            |                         |
            ------------------------------       ---------------------
            |                    |       |       |    |    |    |    |
            |                    |       |       |    |    |    |    |
            |                    |       |       NM   CN   TM   UN   QU
            |                    |       |
            AN                   PL      IN
            |                    |       |
            |                    |       |
       ----------------     ---------    --------------------
       |              |     |   |   |    |   |   |   |   |   |
       |              |     |   |   |    |   |   |   |   |   |
       HU             AL    CO  NM  MS   NM  MA  CO  QU  MS  BP
       |              |                          |
       |              |                          |
--------------   ------------                    |
|  |   |  |  |   |   |   |  |                    QU
T  NM  CO M  F   NM  CO  M  F
```

FM (form of noun) is used for nominalized adjectives, infinitives, or gerunds to indicate their syntactic derivation. The feature FM is always accompanied by the feature DF (derived form - see below). The values for FM are:

A = adjective
I = infinitive
G = gerund

DF (derived form) is often used with derived nominals, usually deverbative forms such as gerunds and agentive nouns, but also sometimes with nouns derived from adjective stems. It further specifies the form underlying the nominal. The values for DF are:

VI = intransitive verb
VT = transitive verb
VR = reflexive verb
A = adjective

FC (form of complement) is used primarily with deverbative nominals which may take a complement of their own. This complement is usually a noun or a phrase representing the subject or object of the verb from which the noun is derived. Since most nominal complements are prepositional phrases, the values of FC are usually prepositions such as VON, AB, IN, etc. The value lambda (LA) is used to indicate absence of a preposition in the complement phrase.

TC (type of complement) is the semantic type of a complement used with a deverbative nominal. The values are those of TY (type) as given above.

TO (type of object) represents the semo-syntactic form of objects used with deverbative nouns. Like TC, the values may be any of those listed under TY above. In addition, the following values are possible:

MI = marked infinitive
TH = "that" clause
GR = gerund
DIR = direction

OB (object) is a syntactic feature used to indicate the case of a nominal used as an object of another nominal. The values include the expected case values associated with objects as well as a value representing objects which are clauses:

A = accusative
D = dative
G = genitive
CL = clause

(Nominative is not included since objects would not be expected to occur in the nominative case.)

IO (interpretation of object) is not a frequent feature with nominals, although it is used with most verbs. The feature is used to interpret the potential objects which a verb (or in this instance, deverbative nominal) may take. The values are:

- O = first object
- O2 = second object
- O4 = reflexive object
- LA = lambda (null)

CAN (canonization) is used for instances in which the derived noun differs from the verb in such a way that none of the conjugated verb forms are identical with it. For example, "destruction" from "destroy" in English. The values are the canonical form of the verb from which the nominal is derived.

TT (tantum noun) is used to indicate nominals which are inherently singular or plural with respect to number agreement.

- S = noun is inherently singular ("Gesundheit ist")
- P = noun is inherently plural ("Leute sind")
- LA = lambda (expresses optionality, e.g., "politics IS" or "politics ARE")

RA (required adverb) is used with nouns which require an adverb modifier. The values represent the semantic class of the adverb. Most such nouns coded thus far take directional adverbs, although the option for other values is not excluded. The values are thus:

- DIR = direction
- LA = lambda (unspecified)

OA (optional adverb) is similar to RA in that it is used to indicate adverbials which may be associated with a nominal. The feature OA is used when the adverb is customary, but optional rather than required. Otherwise the values and uses are the same as RA.

LC (lexical collocation) is used as a subscript for those nouns which typically occur in idiomatic or quasi-idiomatic expressions. The values are:

N = noun only
NP = determiner and/or adjective + noun
PN = preposition + noun
PP = preposition + determiner (+adjective) + noun

LG (source language) is used to indicate the language source for a loan word. This feature is not commonly used in the existing rules, but has been tested in certain examples as a means for identifying foreign words or phrases which do not follow expected morphological patterns. Values which have been supplied for texts studied thus far include:

GR = Greek
FR = French
L = Latin
IT = Italian
SP = Spanish

FR (future research) has been introduced as a feature to mark the need for further study in some of the noun rules. The subscript was created to accommodate unusual nominals in which an adjective + noun combination does not have feature agreement, as in the following example:

Ein Metall in reinem Zustand
= das Metall ist rein
= das reine Metall
* der Zustand ist rein

A set of values has not yet been established.

CP (capitalization) is a feature used to mark the few forms which function as nominals in German but which are usually not capitalized. For example, German "sek.", the abbreviation for "Sekunde" is not capitalized even though the unabbreviated form is. The usual value associated with this feature is N (non-capitalized).

It is anticipated that additional features will be added to the system in the future to account for distinctions related to area of discourse. (Cf. Appendix for a listing of the provenience tags derived from two major German-English dictionaries.) These will serve to more precisely determine rule applications when the system is applied to larger and less restricted corpora.

a. German Paradigmatic Noun Classes

The paradigmatic classification of German nouns and nominalized adjectives represented by the subscript +CL( ) is given in the following tables. The leftmost column is the code number identifying that paradigm. The center column(s) present the four declensional affixes (nominative, genitive, dative, and accusative) for that class. Ø stands for zero inflection; and -O for the ending -o as in "Libretto". The rightmost column shows examples of each class.

Class	Endings	Stems
1	Ø	Luxus
	Ø	Abkehr
	Ø	Boegen
	Ø	
2	Ø	Garten
	-S	
	Ø	
	Ø	
3	Ø	Leute
	Ø	Abwaesser
	-N	
	Ø	
4	Ø	Mann
	-ES/-S	Ball
	Ø/-E	Buch
	Ø	
5	Ø	Haus
	-ES	
	Ø/-E	
	Ø	
6	Ø	Rotliegende
	-N	
	-N	
	Ø	
7	-E	Sael
	-E	Absaetz
	-EN	
	-E	

8	-ER	Buech
	-ER	Laend
	-ERN	
	-ER	
9	-IEN	Antezedenz
	-IEN	
	-IEN	
	-IEN	
10	-EN	Baut
	-EN	
	-EN	
	-EN	
11	-ES	Kodic
	-ES	
	-ES	
	-ES	

	(singular)	(plural)	
12	0	0	Aries, Bizeps
	0	0	Spezies
	0	0	Korps
	0	0	
13	0	0	Alkoven
	-S	0	Zeichen
	0	0	
	0	0	
14	0	-E	Kodex
	0	-E	Graeting
	0	-EN	
	0	-E	
15	0	-E	Amboss
	-ES	-E	Kreuz
	0/-E	-EN	
	0	-E	
16	0	-E	Findling
	-S	-E	Asyl
	0	-EN	
	0	-E	

17	0	-E	Blick
	-S/-ES	-E	Gestuehl
	0/-E	-EN	
	0	-E	
18	0	-E	Filia
	0	-E	
	0	-E	
	0	-E	
19	0	0	Maurer
	-S	0	Geier
	0	-N	Mittel
	0	0	
20	0	-EN	Tat
	0	-EN	Abbildung
	0	-EN	
	0	-EN	
21	0	-EN	Sektor
	-S	-EN	Anion
	0	-EN	
	0	-EN	
22	0	-EN	Mast
	-S/-ES	-EN	Hemd
	0/-E	-EN	
	0	-EN	
23	0	-EN	Herz
	-ENS	-EN	
	-EN	-EN	
	0	-EN	
24	0	-EN	Schmerz
	-ES	-EN	
	0/-E	-EN	
	0	-EN	
25	-E	-EN	Will
	-ENS	-EN	
	-EN	-EN	
	-EN	-EN	

26	Ø	-EN	Zar
	-EN	-EN	
	-EN	-EN	
	-EN	-EN	
27	-E	-EN	Birn(e)
	-E	-EN	
	-E	-EN	
	-E	-EN	
28	Ø	-EN	Herr
	-N/-EN	-EN	
	-N/-EN	-EN	
	-N/-EN	-EN	
29	Ø	-SE	Atlas
	-SES	-SE	Buendnis
	Ø	-SEN	
	Ø	-SE	
30	Ø	-SE	Albatros
	Ø	-SE	Ananas
	Ø	-SEN	
	Ø	-SE	
31	-E	-IEN	Marginal(e)
	-E	-IEN	
	-E	-IEN	
	-E	-IEN	
32	Ø	-IEN	Adverb
	-S	-IEN	
	Ø	-IEN	
	Ø	-IEN	
33	Ø	-ER	Geist
	-S/-ES	-ER	Geld
	Ø/-E	-ERN	
	Ø	-ER	
34	Ø	-N	Blume
	Ø	-N	Abrede
	Ø	-N	
	Ø	-N	

35	Ø	-N	Hasenjunge
	-N	-N	
	-N	-N	
	Ø	-N	
36	Ø/-N	-N	Friede
	-NS	-N	
	-N	-N	
	-N	-N	
37	Ø	-N	Nachbar
	-S/-N	-N	
	Ø/-N	-N	
	Ø/-N	-N	
38	Ø	-N	Genosse
	-N	-N	Eingeborene
	-N	-N	
	-N	-N	
39	Ø	-N	Gevatter
	-S	-N	Auge
	Ø	-N	
	Ø	-N	
40	Ø	-NEN	Abenteuerin
	Ø	-NEN	
	Ø	-NEN	
	Ø	-NEN	
41	Ø	-NEN	Embryo
	-S	-NEN	
	Ø	-NEN	
	Ø	-NEN	
42	Ø	-S	Levee
	Ø	-S	
	Ø	-S	
	Ø	-S	
43	Ø	-S	Clown
	-S	-S	Alibi
	Ø	-S	
	Ø	-S	

44	-A	-EN	Dram(a)
	-AS	-EN	
	-A	-EN	
	-A	-EN	
45	-A	-EN	Lig(a)
	-A	-EN	
	-A	-EN	
	-A	-EN	
46	-O	-I	Librett(o)
	-OS	-I	
	-O	-I	
	-O	-I	
47	-OS	-EN	Kosm(os)
	-OS	-EN	
	-OS	-EN	
	-OS	-EN	
48	-ON	-EN	Kymographi(on)
	-ONS	-EN	
	-ON	-EN	
	-ON	-EN	
49	-ON	-A	Lexik(on)
	-ONS	-A	
	-ON	-A	
	-ON	-A	
50	-UM	-A	Minim(um)
	-UMS	-A	
	-UM	-A	
	-UM	-A	
51	-UM	-EN	Atri(um)
	-UMS	-EN	
	-UM	-EN	
	-UM	-EN	
52	Ø	-TA	Lemma
	-S	-TA	
	Ø	-TA	
	Ø	-TA	

53	-O	-EN	Kont(o)
	-OS	-EN	
	-O	-EN	
	-O	-EN	
54	-US	-ER	Physik(us)
	-US	-ER	
	-US	-ERN	
	-US	-ER	
55	-X	-ZEN	Matri(x)
	-X	-ZEN	
	-X	-ZEN	
	-X	-ZEN	
56	-IS	-EN	Amaryll(is)
	-IS	-EN	
	-IS	-EN	
	-IS	-EN	
57	-IS	-ES	Tenu(is)
	-IS	-ES	
	-IS	-ES	
	-IS	-ES	
58	-US	-I	Mod(us)
	-US	-I	
	-US	-I	
	-US	-I	
59	-US	-EN	Nunzi(us)
	-US	-EN	
	-US	-EN	
	-US	-EN	
60	-E	-I	Monsignor(e)
	-ES	-I	
	-E	-I	
	-E	-I	
61	-EN	-INA	Pronom(en)
	-ENS	-INA	
	-EN	-INA	
	-EN	-INA	

62	-ENS	-ENZIEN	Ag(ens)
	-ENS	-ENZIEN	Akzid(ens)
	-ENS	-ENZIEN	
	-ENS	-ENZIEN	
63	-A	-E	Lir(a)
	-A	-E	
	-A	-E	
	-A	-E	
64	-Y	-IES	Penn(y)
	-Y	-IES	
	-Y	-IES	
	-Y	-IES	

2. German Determiner Features (Category Symbols = D or DET)

The most efficient translation of determiners from German to English requires a separation of many German determiners into the appropriate determiner stem (represented by the symbol D) and a suffix. Thus such pronominal determiners such as "sein" ("his"), which may be used either as pronouns or as pronominal determiners, appear in the lexicon under the symbol D, and are combined with appropriate suffixes to indicate case, gender, and number. Other determiners such as "das" or "derjenige" are entered in the lexicon as DET and are included in their inflected forms.

The following subscripts are used in the description of determiners in the German lexicon:

CL	=	class
FM	=	form
CA or C	=	case
NU or N	=	number
GD or G	=	gender
IN	=	inflection
TY	=	type
WD	=	word
BF	=	bound form
SNC	=	syncopated form

CL (class) is used only with stems of "D" determiners. It represents morphological class and is used to ensure agreement between the stem and affix used. "CL" in determiners, as in other constituents, takes a numerical value. Two classes are specified, according to whether or not an affix is required.

1 = may stand alone ("kein", "unser")
2 = cannot occur without an affix ("jen", "dies")

FM (form) is used with both "D" and "DET" determiners. It represents the manner in which the determiner is used. A German word which can function either as a determiner or as a pronoun, depending upon its environment, is listed in the lexicon only once, thus eliminating a source of multiple analyses. The feature FM is used to indicate the potential functions of such words. (Cf. the discussion of German determiners and pronouns at the end of this sub-section.) The values of FM include:

DET = determiner
DEM+P = demonstrative pronoun
REL+P = relative pronoun
IND+P = indefinite pronoun
INT+P = interrogative pronoun

CA (case) has the expected values one associates with German grammar, namely:

N = nominative
G = genitive
D = dative
A = accusative

NU (number) refers to grammatical number, with the values:

S = singular
P = plural

GD (gender) represents grammatical gender and has the values:

M = masculine
F = feminine
N = neuter

(The subscripts G and N have also been used with certain determiners such as "deren" and "dessen" in some forms of the dictionary. The values are the same as for GD and NU.)

IN (inflection) indicates whether the form takes strong or weak endings in German morphology:

S = strong inflection
W = weak inflection

TY (type) is used with determiners which may function as pronouns. It represents the potential semantic class of the referent, and thus may take the same values as TY does with nouns. (See the section on nouns for a fuller description of possible values.)

WD (word) has as its value an abbreviation for the determiner itself. It is used for determiners which are listed in the lexicon in several inflected forms, but which at some time in analysis must be identified as being the same root form. For example, only the item "was" may be used as a relative pronoun modifying a clause rather than a noun or noun phrase. For this reason, it contains the feature WD(W).

BF (bound form) is used for determiners (pronouns) which may occur as the second element in a contracted form with a preposition. The value of BF is the string itself, for example M (as in "im") or R (as in "zur").

SNC (syncope) is used in those determiners which are syncopated forms, such as "unsr-". The value given SNC is the deleted letter.

a) German Determiner/Pronoun Overlap

As mentioned earlier, German dictionary items which may function as either determiner or pronoun, depending on their environment, are coded only once, with a complex label which contains their features as determiners and their features as pronouns. (They are identified as possible determiners by the value DET under the subscript FM.) This prevents multiple analyses of such items regardless of their environment, a considerable savings because of their frequency of occurrence in actual texts. Examples are shown below from the LRC German dictionary.

```
C 198        V  DET             *  DENSELBEN
D 4          +  GD(M'M,F,N)/    P
             +  NU(S'P)/
             +  CA(A'D)
             +  IN(S)
             +  FM(DET,P+DEM)
             +  TY(HU,AL,PL,
                IN,NT,AB)
             \
```

```
C 149          V  DET              *  DER
D 1            +  GD(M'F'M,F,      P
                  N)/
               +  NU(S'S'P)/
               +  CA(N'G,D'G')
               +  IN(S)
               +  FM(DET,P+DEM,
                  P+REL)
               +  G(M'F)/
               +  C(N'D)
               +  N(S)
               +  TY(HU,AL,PL,
                  IN,NT,AB)
               \
               T  1.5

C 197          V  DET              *  DERJENIGE
D 1            +  GD(M)            P
               +  NU(S)
               +  CA(N)
               +  IN(S)
               +  FM(DET,P+DEM)
               +  TY(HU,AL,PL,
                  IN,NT,AB)
               T  1.5
               \
```

Rule C 149.1 classifies the item "der" as determiner, demonstrative pronoun, and relative pronoun.

Relative pronouns are given the subscripts G, C, and N, in addition to GD, CA, and NU. The latter set of features is for agreement with the following nominal in relative noun phrases:

> "Es geschah in 1963, zu welcher Zeit derartiges noch ganz unerwartet war."

The features G and N must agree with the gender and number of the preceding nominal which is being modified by the relative clause. C (case) must contain the case governed by the verb:

> "die Explosion, deren man sich noch heute erinnert."

3. German Pronoun Features (Category Symbol = PRN)

As pronouns in German are inflected for case, number, person, and gender, such features are specified in their description. Inclusion of two other semantic and syntactic characteristics has also proven useful in attaining quality translation. The pronoun features used at present are:

PS = person
CA = case
NU = number
GD = gender
FM = form class
TY = semantic type

The rules which analyze pronoun as NP's assign the gender, number, case, and semantic type features of the pronoun to the NP. For personal pronouns, the PS feature is also assigned to the NP; for all other pronouns, the feature PS(3) marks the NP as 3rd person.

PS (person) is for grammatical person.

1 = first person
2 = second person
3 = third person

CA (case) refers to grammatical case. The values are the four used in German, namely:

N = nominative
G = genitive
D = dative
A = accusative

NU (number) refers to the grammatical number distinctions made in the language. German uses two values:

S = singular
P = plural

GD (gender) is used to indicate grammatical gender, with respect to which, pronouns and related constituents must be in agreement.

M = masculine
F = feminine
N = neuter

FM (form) categorizes the class of pronoun in question. This feature also occurs with stems which may function either as determiners in relation to nouns, or alone as pronouns. The values are:

DEM+P = demonstrative pronoun
REL+P = relative pronoun
IND+P = indefinite pronoun
INT+P = interrogative pronoun
POSS+P = possessive pronoun
PERS+P = personal pronoun
REF+P = reflexive pronoun
REC+P = reciprocal pronoun

TY (type) represents the semantic class of the referent of the pronoun. The values are identical to the values permitted with nouns. (Cf. the noun TYPE list.)

4. German Adjective Features (Category Symbol = A)

German adjective stems may vary in terms of the number of syntactic and semantic features required for translation. Simple adjectives without multiple ambiguities or complex selectional restrictions requiring more than one English translation may carry as few as two features, namely CL (morphological class) and TM (type of modificand), since these are required to ensure correct parsing of any German string. A variety of additional features may also be utilized, however.

The semantic and syntactic features which have been used thus far in the LRC dictionaries for German adjective stems include:

- CL = morphological class
- TM = semantic type of modificand
- FM = syntactic form of modificand
- TY = type of adjective stem
- SP = participial form
- FO = syntactic form of object
- TO = semantic type of object
- IO = interpretation of object
- LC = lexical collocation

CL (morphological class) is used to indicate the paradigmatic class of the adjective stem. There are twenty adjective classes in use in the system at present, as listed in the chart on the following page. (The Ø represents a zero morpheme; a wavy line signifies the non-occurrence of a form at that point in the paradigm.)

PARADIGMATIC ENDINGS

CLASS NAME	Predi- cative	Compar- ative	Superla- tive	EXAMPLES
1	-E/0	-ER	-ST/-EST	stupid, sproed
2	"	"	-ST	aeusser, ober
3	0	-ER	-EST	frevelhaft
4	0	-ER	-ST/-EST	zaeh
5	0	-ER	-ST	fein, gruen
6	0/-E	-ER	-EST	weis, mued
7	"	-ER	"	dunkl, edl
8	"	"	"	hoh
9	0/-E	-ER	-ST	feig
10	0/-E	"	"	nah, bang
11	0	"	"	gross
12	0	"	"	hoch
13	"	-ER	-EST	kuerz
14	"	-ER	-ST	schaerf
15	"	-ER	"	hoeh, naeh
16	"	"	-ST	hoech, naech
17	"	-ER	-T	groess
18	0	"	-ST	dunkel, edel
19	0	"	"	rosa
20	"	"	"	Muenchner (also abbreviations)

TM (semantic type of modificand) is used to indicate the required semantic type of any noun which can be modified by the adjective. For example, an adjective stem such as "optimal" would require an abstract modificand. Thus, the values for TM are the same values as those associated with TY for nouns, and a complete description of the value tree may be found in the noun section, above.

FM (syntactic form of the modificand) is used to indicate the constituent classes which may be modified by the adjective. Depending upon whether the adjective modifies a noun or a clause, the values may be:

NO = noun
CL = clause

TY (type of stem) is usually used to indicate adjectives which have multiple stem forms, for example "hoh" and "hoeh". The associated value is typically MSR, indicating recognition of multiple stems.

SP (participial form) is used to indicate a deverbative adjective.

PAPL = past participle
PRPL = present participle

The last three adjective features are similar to verb features and are usually specified for all deverbative adjectives, i.e., present or past participles used as adjectives. Since deverbative adjectives may occur with objects just as the verb stem may, it is necessary to include those features describing verb-object relations with deverbative adjectives. The features are also used with other adjectives which may have an associated object but are not necessarily participial in derivation. For example, "unabhaengig" may take a prepositional phrase with "von", as in English "heedless" might take "of".

FO (form of object) has as values the syntactic form of the constituent which functions as object of a deverbative adjective. In case of a prepositional object, the value would be the preposition itself. Typical values would be:

LA = lambda (null)
CL = clause
VON = "von" phrase
AB = "ab" phrase
FUER = "fuer" phrase
etc.

TO (type of object) is used to indicate the semantic type of a potential object. The values are the same as those which are used for TY with nouns, and so are not repeated here.

IO (interpretation of object), which serves to enumerate the various objects which may occur with a verb, may also be used with deverbative adjectives in the same manner. The values are:

O	=	first (direct) object
O2	=	second (indirect) object
O4	=	reflexive object
LA	=	lambda (no object or optional object)

LC (lexical collocation) marks adjectives which may participate in an idiomatic or quasi-idiomatic phrase. The values reflect the structure of the phrase.

5. German Verb Features (Category Symbol = V)

The most complex of the syntactic classes which must be considered in any MT system are the verbs. They must agree with their subjects with respect to number and semantic class, and with their objects with respect to case, number, and semantic class. Verbs also carry markers for tense, mood, and voice. Since a predicate may consist of more that one verbal element, it is necessary to distinguish between finite verb forms (those which carry tense and number markers) and non-finite verb forms (such as infinitives and gerunds).

In the LRC MT system the distinction is made between verbs which may function as either finite or non-finite forms, and those which may only be finite. The latter category, consisting of those verbs which have traditionally been considered modals, will be considered in the next section.

The complete list of subscripts used thus far for non-modal verbs is:

CL = inflectional class
PX = prefixes
FS = form of subject
TS = type of subject
FO = form of object
TO = type of object
IO = interpretation of object
IS = interpretation of subject
RA = required adverb
OA = optional adverb
AX = form of auxiliary
LC = lexical collocation

CL (class) is used to specify the classes of inflectional affix which a verb may take. The values are indicated by numbers, as is true of CL with most other constituents. A chart of the 67 verb classes is shown at the end of this sub-section.

PX (prefix) marks the many German verbs that can occur with preposed affixes. It is necessary to enumerate which prefixes may be used with such verbs, and numerous values occur since each prefix has its own. Most of the values are identical to

the surface form of the prefix in question, for example AUF, HERAB, FERN, AUS, etc. The value LA is also used to indicate a null value or optionality.

FS (form of subject) distinguishes between the two syntactic structures which may function as subjects of a verb.

N = noun
CL = clause

TS (type of subject) represents the semantic class to which the subject associated with the verb must belong. For example, a verb such as "denken" must have an animate subject. The values are thus the same as those associated with TY (type) in nouns, q.v.

FO (form of object) is used to stipulate the syntactic form of any constituents which may function as objects of a verb. A wide variety of values may occur. Some verbs such as "glaub" may take a clause as object. Others require a noun phrase in a particular case, generally accusative or dative. In addition, German has verbs which take prepositional objects, for example "gelt". The values usually encountered include:

CL = clause
A = accusative nominal
D = dative nominal
ALS = "als" phrase
FUER = "fuer" phrase
ZU = "zu" phrase
etc.

The value LA (lambda) is also used to indicate optionality.

TO (type of object) is similar to TS in that it represents the semantic class of an associated nominal, in this instance the object. The values are the same as TY in nouns, q.v.

IO (interpretation of object) is used to indicate in part the functional relationship between a potential object and its corresponding verb. Values indicate an object or adverbial function:

O = first (i.e., direct) object
O2 = second (i.e., indirect) object
O3 = potential third object (not usually required)
O4 = reflexive object
1 = first adverbial predicate
2 = second adverbial predicate
3 = third adverbial predicate
LA = lambda (no predicate, or optionality)

IS (interpretation of subject) is obligatory with verbs, although it usually simply indicates that the verb takes a subject. However, it is still necessary since verbs which may occur without a nominative do exist, for example "geling" or "gelt". The values are thus:

S = expressed subject
LA = no expressed subject

RA (required adverb) marks verbs which obligatorily take an adverbial. Examples would include certain uses of "geh" which require a directional adverb, or uses of "seh" with a manner adverb. The usual values are:

MAN = manner adverb
DIR = directional adverb
LA = no adverb
L+ST = locative and stative
T+DU = time and durative

OA (optional adverb) is similar to RA except that its values are not obligatory.

MAN = manner adverb
DIR = directional adverb
LA = unexpressed

AX (form of auxiliary) specifies whether the verb takes a form of "haben" or of "sein" as its auxiliary.

H = haben
S = sein
LA = lambda (unspecified)

LC (lexical collocation) is used to denote participation of the verb in an idiomatic expression, usually with set (required) elements. The surface elements which make up the expression may not be contiguous, i.e., there may be discontinuous constituents. The value usually specifies the type of construction with which the verb participates:

- NP = noun and/or adjective + noun phrase
- N = noun only
- PN = preposition + noun
- PP = preposition + determiner (+ adjective) + noun
- A = adjective only

a. Paradigmatic Verb Classes

In the following chart of classes of German verb stems, the far right column shows one or more stems exemplifying that class. The leftmost column gives the number assigned to that stem-class, and in three instances includes an ending required by the infinitive forms in that class. The four central columns list the endings (including 0 for zero inflection) for each paradigmatic form which can occur with that class of stem. A blank indicates the non-occurrence of a stem-class at that point in the paradigm. Where some but not all of the six customary forms occur in the columns for present and past tense, the ones which appear are marked for person and number. In these same columns it may be assumed that the endings shown are indicative unless they fall below a wavy line which signals that the following endings shown for that class are subjunctive. Similarly, in the column for imperatives a wavy line of demarcation separates the ending for a singular from that for the plural imperative. Possible variants in form of an ending are separated by a slash.

CLASS	Present INDIC. / SUBJUNC.	Past INDIC. / SUBJUNC.	Imperative SINGULAR / PLURAL	Past Participle	EXAMPLES
1				-EN	gelung besess
2				GE-EN	gang gess borst
3	-ST 2s -T 3s				baeck
4	-T 2s -T 3s				laess
5	-ST 2s 0 3s				haelt
6	-ST 2s -T 3s		0 ---		sieh befiehl

7	-T 2s		Ø	iss
	-T 3s		~~~	
8	-ST 2s		Ø	gilt
	Ø 3s		~~~	ficht
9	-EST/-ST 2s		Ø	drisch
	-T 3s		~~~	erlisch
10	Ø 1s			vermag
	-ST 2s			bedarf
	Ø 3s			
11	-E 1s		-E	handl
	~~		~~~	
	-E 1s			
	-E 3s			
12		Ø		sah
		-ST		buk
		Ø		
		-EN		
		-T		
		-EN		
13		Ø		besass
		Ø/-EST		
		Ø		
		-EN		
		-T		
		-EN		
14	Ø			fand
	-ST/-EST			
	Ø			
	-EN			
	-ET			
	-EN			
15	~~~~			
	-E			goess
	-EST			beschoess
	-E			
	-EN			
	-ET			
	-EN			

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
16	~~~~				
	-E				schwuer
	-ST/-EST				saeh
	-E				buek
	-EN				
	-T/-EN				
	-EN				
17	~~~~				
	-TE				daech
	-TEST				haet
	-TE				
	-TEN				
	-TET				
	-TEN				
18	0			GE-EN	
	-ST/-EST				trog
	0				
	-EN				
	-T				
	-EN				
19	0			-EN	betrog
	-ST/-EST				erlosch
	0				
	-EN				
	-T				
	-EN				
20	-TE			-T	erkann
	-TEST				
	-TE				
	-TEN				
	-TET				
	-TEN				
21	0			-EN	bestand
	-ST/-EST				
	0				
	-EN				
	-ET				
	-EN				

22	Ø		-EN	erschloss
	-ES			
	Ø			
	-EN			
	-T			
	-EN			
23	-TE		GE-T	brann
	-TEST			konn
	-TE			
	-TEN			
	-TET			
	-TEN			
24	Ø		GE-EN	stand
	-EST			
	Ø			
	-EN			
	-ET			
	-EN			
25	Ø		GE-EN	floss
	-EST			
	Ø			
	-EN			
	-T			
	-EN			
26		-E		vermoeg
		----		wiss
		-T		
	-EN 1p			
	-T 2p			
	-EN 3p			

	-E			
	-EST			
	-E			
	-EN			
	-ET			
	-EN			

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
27	-E		----		ess
			-T		brech
	-EN				
	-T				
	-EN				

	-E				
	-EST				
	-E				
	-EN				
	-ET				
	-EN				
28	-E		----		fecht
			-ET		
	-EN				
	-ET				
	-EN				

	-E				
	-EST				
	-E				
	-EN				
	-ET				
	-EN				
29	-E		Ø/-E		besauf

			-T		
	-EN				
	-T				
	-EN				

	-E				
	-EST				
	-E				
	-EN				
	-ET				
	-EN				

30 -E ---- -EN umgeb
-T

-EN
-T
-EN

-E
-EST
-E
-EN
-ET
-EN

31 -E ---- GE-EN seh
-T

-EN
-T
-EN

-E
-EST
-E
-EN
-ET
-EN

32 -E ---- -EN betret
-ET

-EN
-ET
-EN

-E
-EST
-E
-EN
-ET
-EN

CLASS	Present INDIC. -------- SUBJUNC.	Past INDIC. -------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
33		Ø -ST Ø -EN -T -EN ---- -E -EST -E -EN -ET -EN			ging
34		Ø -T/-EST Ø -EN -T -EN ----- -E -EST -E -EN -ET -EN			blies
35		Ø -EST Ø -EN -ET -EN ----- -E -EST -E -EN -ET -EN			erriet

```
..........................................................................
36       -E                        0/-E    -EN      uebertrag
                                   -----

                                   -T
         -EN
         -T
         -EN
         ----
         -E
         -EST
         -E
         -EN
         -ET
         -EN
..........................................................................
37       -E                        0/-E     GE-EN   back
                                   -----

                                   -T
         -EN
         -T
         -EN
         ----
         -E
         -EST
         -E
         -EN
         -ET
         -EN
..........................................................................
38       -E                        0/-E     -EN     gerat
                                   -----

                                   -ET
         -EN
         -ET
         -EN
         ----
         -E
         -EST
         -E
         -EN
         -ET
         -EN
..........................................................................
```

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
39	-E -EN -ET -EN ---- -E -EST -E -EN -ET -EN		Ø/-E ----- -ET	GE-EN	brat
40		Ø -ST Ø -EN -T -EN ---- -E -EST -E -EN -ET -EN		-EN	erblich
41		Ø -ST Ø -EN -ET -EN ---- -E -EST -E -EN -ET -EN		-EN	vermied

42	Ø	GE-EN	blieb
	-ST		
	Ø		
	-EN		
	-T		
	-EN		
	~~		
	-E		
	-EST		
	-E		
	-EN		
	-ET		
	-EN		
43	Ø	-EN	entriss
	-EST		
	Ø		
	-EN		
	-T		
	-EN		
	~~		
	-E		
	-EST		
	-E		
	-EN		
	-ET		
	-EN		
44	Ø	GE-EN	glitt
	-ST		
	Ø		
	-EN		
	-ET		
	-EN		
	~~		
	-E		
	-EST		
	-E		
	-EN		
	-ET		
	-EN		

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti-ciple	EXAMPLES
45		Ø -ST Ø -EN -T -EN ~~~~ -E -EST -E -EN -ET -EN		GE-EN/-N	schrie
46		Ø -EST Ø -EN -T -EN ~~~~ -E -EST -E -EN -ET -EN		GE-EN	riss
47		-TE -TEST -TE -TEN -TET -TEN ~~~~~ -TE -TEST -TE -TEN -TET -TEN		-T	ueberrann

48	-E -ST -T -EN -T -EN ~~~~ -E -EST -E -EN -ET -EN	Ø/-E ~~~~~ -T	steh geling
49	-E -EST -ET -EN -ET -EN ~~~~ -E -EST -E -EN -ET -EN	Ø/-E ~~~~~ -ET	bind gleit
50	-E -T -T -T -EN -T -EN ~~~~ -E -EST -E -EN -ET -EN	Ø/-E ~~~~~ -T	giess heiss genes

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti-ciple	EXAMPLES
51 -N	-E -ST -T -N -T -N ---- -E -EST -E -EN -ET -EN		Ø/-E ----- -T		tu vertu
52	-E -ST -T -EN -T -EN ---- -E -EST -E -EN -ET -EN		Ø/-E ----- -T	-EN	beruf
53	-E -ST -T -EN -T -EN ---- -E -EST -E -EN -ET -EN		Ø/-E ----- -T	GE-EN	komm

54	-E		-E		gebaer
			----		erloesch
			-T		
	-EN				
	-T				
	-EN				
	----	----			
	-E	-E			
	-EST	-EST			
	-E	-E			
	-EN	-EN			
	-ET	-ET			
	-EN	-EN			
55	-E		Ø/-E		erkenn
	-ST		-----		brenn
	-T		-T		
	-EN				
	-T				
	-EN				
	----	-----			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-EN	-TEN			
	-ET	-TET			
	-EN	-TEN			
56	-E	-TE	-E	-T	besag
	-ST	-TEST	----		
	-T	-TE	-T		
	-EN	-TEN			
	-T	-TET			
	-EN	-TEN			
	----	-----			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-EN	-TEN			
	-ET	-TET			
	-EN	-TEN			

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
57	-E	-TE	-E	GE-T	waehl
	-ST	-TEST	~~~~		
	-T	-TE	-T		
	-EN	-TEN			
	-T	-TET			
	-EN	-TEN			
	~~	~~~~~			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-EN	-TEN			
	-ET	-TET			
	-EN	-TEN			
58	-E	-TE	-E	GE-T	abenteuer
-N	-ST	-TEST	~~~~		
	-T	-TE	~~~~		
	-N	-TEN			
	-T	-TET			
	-N	-TEN			
	~~	~~~~~			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-N	-TEN			
	-T	-TET			
	-N	-TEN			
59	-E	-TE	-E	-T	verwunder
-N	-ST	-TEST	~~~~		
	-T	-TE	-T		
	-N	-TEN			
	-T	-TET			
	-N	-TEN			
	~~	~~~~			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-N	-TEN			
	-T	-TET			
	-N	-TEN			

60	-E	-ETE	-E	GE-ET	send
	-EST	-ETEST			antwort
	-ET	-ETE	-ET		acht
	-EN	-ETEN			
	-ET	-ETET			
	-EN	-ETEN			
	-E	-ETE			
	-EST	-ETEST			
	-E	-ETE			
	-EN	-ETEN			
	-ET	-ETET			
	-EN	-ETEN			
61	-E	-TE	-E	GE-T	aechz
	-T/-EST	-TEST			
	-T	-TE	-T		
	-EN	-TEN			
	-T	-TET			
	-EN	-TEN			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-EN	-TEN			
	-ET	-TET			
	-EN	-TEN			
62	-E	-ETE	-E	-ET	beobacht
	-EST	-ETEST			
	-ET	-ETE	-ET		
	-EN	-ETEN			
	-ET	-ETET			
	-EN	-ETEN			
	-E	-ETE			
	-EST	-ETEST			
	-E	-ETE			
	-EN	-ETEN			
	-ET	-ETET			
	-EN	-ETEN			

CLASS	Present INDIC. ------- SUBJUNC.	Past INDIC. ------- SUBJUNC.	Imperative SINGULAR -------- PLURAL	Past Parti- ciple	EXAMPLES
63	-E	-TE	-E	-T	durchsetz
	-T/-EST	-TEST	----		
	-T	-TE	-T		
	-EN	-TEN			
	-T	-TET			
	-EN	-TEN			
	-----	-----			
	-E	-TE			
	-EST	-TEST			
	-E	-TE			
	-EN	-TEN			
	-ET	-TET			
	-EN	-TEN			
64	-E				moecht
	-EST				
	-E				
	-EN				
	-ET				
	-EN				

	-E				
	-EST				
	-E				
	-EN				
	-ET				
	-EN				
65	Ø 2s		Ø		birst
	Ø 3s				
66		Ø			barst
		-EST			
		Ø			
		-EN			
		-ET			
		-EN			
67	Ø 1s				weiss
	-T 2s				
	Ø 3s				

6. German Modal Features (Category Symbol = FIN)

German has a variety of auxiliaries such as "haben", "moegen", "sein", "lassen", etc. The features used with this class include:

TN	=	tense
PS	=	person
NU	=	number
MD	=	mood
FS	=	form of subject
TS	=	type of subject
IS	=	interpretation of subject
FO	=	form of object
TO	=	type of object
IO	=	interpretation of object
TY	=	type of modal
WD	=	word group
FM	=	form of modal
AX	=	form of auxiliary
LC	=	lexical collocation
RA	=	required adverb

Many of the features are the same as those found with the non-finite verbs of the previous section. However, additional information is specified for the modals since they appear in the lexicon in their inflected forms, whereas most other verbs are entered as stem forms and derive tense, mood, and number information from their suffixes.

TN (tense) indicates the tense of the verb form. Since German has two sets of inflectional suffixes for tense, there are two possible values:

PR = present
PA = past

PS (person) specifies the distinctions of grammatical person.

1 = first person
2 = second person
3 = third person

NU (number), as one might expect, has two possible values:

S = singular
P = plural

MD (mood) for German has the associated values:

I = indicative
S = subjunctive

FS (form of subject),
TS (type of subject),
IS (interpretation of subject),
FO (form of object),
TO (type of object), and
IO (interpretation of object) are used to indicate the grammatical form, semantic type, and sentential function of constituents which may be used as subject or object of the verb in question. Since the values possible with modals are the same as those used with other verbs, the values will not be repeated here. (A complete list may be found in the immediately previous section on verb features.)

TY (type of modal) indicates potential syntactic usage.

M = modal
V = verb (i.e., as the main verb of the clause)
C = copula
W+A = "werden" phrase
S+A = "sein" phrase
SZ = used with "sein"

WD (word group) is introduced to specify the stem form of the modal in question, since the actual orthographic representations may vary considerably with changes in tense, mood, person, and number. The values are:

S = sein
D = duerfen
H = haben
K = koennen
MG = moegen
MS = muessen
SL = sollen
W = werden
WL = wollen

FM (form of modal) is used with non-finite forms of the modal verbs as an indication of grammatical usage. The values are:

INF = infinitive
PAPL = past participle

AX (form of auxiliary) expresses the class of finite verbs which may be used with a non-finite form of the modal. The values are:

H = haben
S = sein
M = another modal

LC (lexical collocation) signifies that the auxiliary may participate with an idiomatic or quasi-idiomatic expression. Values which have been used with modals for this purpose include:

A = adjective only
PP = preposition + determiner (+ adjective) + noun
LA = lambda

RA (required adverb) and NOPX (no prefix) were at one time used with the modal verbs. The subscripts were used to differentiate between the modals and other verbs in certain syntactic rules, but are no longer in use in the LISP version of the system. The value, where the features were used, was always LA (lambda).

7. German Prefix Features (Category Symbol = PRFX)

This category is primarily for those elements which may be prefixed to German verbs. Since the co-occurrence restrictions between prefixes and verbs are usually included in the verb description (i.e., each verb lists its possible prefixes in its features), it is only necessary to assign values such that a particular prefix may be identified. Thus most German prefixes only have the one subscript PX, which has the prefix string itself as its value.

8. German Preposition Features (Category Symbol = PREP)

The following subscript features are used for prepositions:

TYP = type
GC = governs case
TO = type of object
PR = preposition
CN = contracted form
ON = onset

TYP (type) refers to the syntactic function of the resulting prepositional phrase, i.e., whether it is used as an adverbial or as the object of a verb.

PO = prepositional object
PR = prepositional phrase

GC (grammatical case) specifies which case is required for the object of the preposition.

G = genitive
D = dative
A = accusative

TO (type of object) has the same values as those of noun TY, q.v.

PR (preposition) indicates the particular one being represented. It is generally used with prepositions which have variable forms. For example, VON and VO (as in the contracted form "vom") both have the value VON to indicate that these are forms of the same preposition.

CN (contracted) is used as a subscript with those prepositions which occur in contracted form. The value is that of the consonant which may follow, for example the preposition "zu" has two values for CN: R and M.

ON (onset) indicates the initial element of the preposition. It is employed as a means for indicating the allomorph which occurs in "da" derivative forms since "r" is introduced before a preposition beginning with a vowel (e.g., "darum").

C = consonant
V = vowel

9. German Adverb Features (Category Symbol = AV)

In the present German dictionaries used by the LRC translation system, most adverbials have very few if any associated features since syntactic and semantic distinctions within this word class are usually not required for sentence parsing or disambiguation. There are of course some exceptions, for example "moeglichst":

```
C 40102   V AV           * MOEGLICHST
          + EX(LA)       P
          + DK(A)        F
          + MD(EQ)
          + MGL(LA)
          + NADV(LA)
```

Yet, although complex sets of features are not recorded for most of the entries in the limited lexicon now in use, a large set of potential adverb features has been devised for use when required by the demands of quality translation. These are listed in detail in the description of English adverb features. The features most likely to be in use in the version of the system in use at at the time of the Demonstration (June, 1974) include:

EX = expanded adverb
DK = declension class
TYP = semantic type
MD = modificand
DC = declension category
SP = special usage
LC = lexical collocation
MGL = [see below]
NADV = [see below]

EX (expanded form) has been used as a potential feature for indicating complex grammatical constructions used as adverbs. As such, it does not usually occur in the lexicon, but may be introduced by grammatical rules.

DK is used to indicate the lexical categories an adverb may modify. The values are referred to in analysis to determine the relationship between an adverb and other sentence constituents. The values associated with DK are:

N = noun phrase
ADV = adverbial
PN = post-posed noun phrase
PAV = post-posed adverbial
A = adjective
NA = nominal adjective
VB = verb
NUM = number

TYP (semantic type) is used to identify an adverb according to semantic class. The usual values are:

T = temporal (time)
L+ST = locative or status

MD (syntactic form of modificand) specifies the syntactic classes of constituents which an adverb may modify.

P = positive
EQ = equative
CP = comparative
SP = superlative
T = temporal [used with "noch"]

DC (declension category) is similar to DK, but may be used for additional specificity. Values which have been used are:

A = adjective
CL = clause
WIE = "wie" clause

SP (special usage) has been used to indicate such attributes as inherent negation. The value is usually

N = negative

although other values may be introduced in the future.

LC (lexical collocation) is used as with other grammatical categories to indicate forms which are characteristically used in idiomatic expressions.

MGL (moeglich) and NADV (negative adverb) were at one time used to indicate special forms. These features have been replaced in most of the rules by other features of more general applicability.

10. German Conjunction Features (Category Symbol = CONJ)

Conjunctions carry the following subscripts:

CJ = conjunct
KT = clause type
WD = word
CAT = category
TYP = type

CJ (conjunct) represents the grammatical class of the conjoined constituents. It is used to indicate whether a particular conjunction is used only to conjoin clauses, or whether it may be used with lower level predicates as well. The values are:

C = clause
P = predicate

KT (clause type) indicates the function of a constituent introduced by the conjunction. Values are:

C = coordinate
S = subordinate
I = independent

WD (word) was introduced to differentiate between conjunctions with similar features when necessary. The value is an abbreviation for the particular conjunction, for example "oder" has WD(O) and "und" has WD(U).

CAT (category) specifies conjunctions which may be used adverbially, such as "aber", and at present has only the value AV.

TYP (type) is used with "dass" and has the values:

CMPL = complement clause
CONS = consequent clause
FINAL = final

III. GERMAN SURFACE STRUCTURE

Once individual morphemes have been identified by the action of the dictionary programs, the next step is to build the morphemes up into words and the words into sentences. The LRC system was initially conceived with two levels between dictionary and the complete sentence parse, corresponding to the traditional distinction between morphology and syntax. Each of the two analysis stages (referred to as Word Analysis and Syntactic Analysis) were to be followed by a corresponding choice stage for the purpose of checking well-formedness conditions and performing transformations based upon the results of these tests. Since the choice operations depend upon the results of an analysis phase, most choice operations were delayed until its completion. However, the value of choice operations in eliminating inappropriate analyses made it desirable to perform some Syntactic Choice operations before the analysis of the sentence had been completed, thus eliminating many multiple analyses.

A. Word Component

The purpose of the Word Component is fourfold:

- a- to find all segments or sequences of segments which are well-formed according to the word grammar of a language, and record that description;
- b- to delete all segments or sequences of segments which are not well-formed;
- c- to determine those text positions where a sentence or clause boundary could theoretically occur;
- d- to identify the nodes on which syntactic analysis is to continue building.

The reasons for (a) are obvious.

The incorrect segments are removed (b) to reduce processing time and to increase the amount of work space and grammar information that can be kept in store during processing.

Tentative sentence boundaries (c) are introduced for this reason: a direct substitution analyzer without any restrictions will provide two sentence readings for the text "the experiments succeeded": one for "experiments succeeded" and another for "the experiments succeeded". By making clause rules and sentence rules sensitive to potential sentence boundaries, the clause and

sentence interpretations could be restricted. The sequence "the demonstrations and the experiments were performed" would thus show only two analyses as sentences, namely for the span "the experiments were performed" and for the whole text. (These two analyses are not at present reduced to one since the word "and" as well as the "." are used to establish potential sentence boundaries.)

The restrictions of constituents which could be built on by syntactic analysis (d) are obtained by flagging all nodes except for the top node of a constituent, thereby making them 'invisible' to the syntactic analyzer. A sequence like "had been seen" would thus be interpreted as a predicate and none of the the intermediate structures would be available for subsequent syntactic analysis.

1. Word Analysis

Word Analysis applies all rules applicable to every text span available at each position in the text. The application of these rules can be restricted by means of the operators P and F which can be attached to constituents in a rule consequent. The operator P indicates that the constituent must be preceded by a marginal symbol, the operator F stands for 'constituent must be followed by a marginal symbol'. By means of these operators, a large number of non-productive interpretations occurring during the morphological interpretation of a word can be avoided. Thus the span "GROESSE" occurring in the word "GROESSEN" is not interpreted as a nominal (a noun + its inflection endings) since it is not followed by a marginal symbol. Word Analysis operates with the word grammar of a language, whose initial symbol for all languages is WORD. Any text span covered by an analysis dominated by WORD is thus well-formed according to the word grammar of that language.

2. Word Choice

Word Choice performs the functions under (a) through (d) above:

a- by retaining all readings dominated by WORD;

b- by destroying all interpretations not dominated by WORD, including the artificial symbol WORD;

c- by inserting potential sentence boundaries into the text spans according to the specifications of the Word Analysis rules applying to that text span; and

d- by flagging every node which was not immediately dominated by WORD.

In order to permit Word Choice to execute these functions more efficiently, special operators can be attached to word rules.

The D-operator permits the destruction of a node and of every node building on it. It was used to destroy interpretations covering a word which contained internal syntactic boundaries like the German "am" which consists of a preposition followed by a determiner.

The S-operator permits the attachment of 'super flags' to nodes directly dominated by WORD. "Super-flagging" is used to prevent the unflagging of a node.

The I-operator permits the insertion of new subscripts and values into a constituent. It was thus possible to attach information pertaining to the context in which a word occurred to the constituent, and thereby restrict its rewriting during the subsequent Syntactic Analysis phase.

The most powerful operator is the P-operator (preference operator). It can only be attached to the constituent WORD. It permits the algorithm to destroy every constituent in a text span not dominated by this particular symbol WORD. The P-operator is executed according to the longest span principle In other words, a span with a P-operator has precedence over any internal span also constituting a WORD. The latter are deleted.

Word Analysis uses the rules of the German Surface Word Grammar [GFG-W], applying the rules to the output of the dictionary parse, while Syntactic Analysis applies to the output of Word Analysis and Choice. As the word and syntax grammars employ the same rule format conventions, the two grammars may be discussed together.

Both the word and syntactic algorithms first compare the category symbols on the right side of the rule to be applied with the category symbols found in the output of the previous level's operations. These symbols correspond to the labels given nodes in a phrase-structure or transformational grammar. For example, the following rule would apply to a PRFX (prefix), a V (verb stem), and a V-FLEX (verb inflection) created by prior rules and would form a VB (verb).

```
C 84   V VB      V PRFX     V V     V V-FLEX
D 6    $*2.2CX   $ PX       $ CL    ? IX(LA)
       $*2.3FX   . 3.1,4.2  ? NP    $ CL
       ^ 4       . 2.1,3.3  $ PX    B
       ^ 3       P          B

       1
       A 3PX(1FX)
       S 3-5
```

In addition to the category symbol variables, the word and syntax rules may contain dummy symbols. Dummy symbols occur on the top line of the rule and are identified by the symbol D (dummy) rather than a V (variable). The dummy symbol, unlike variable category symbols, does not represent a constituent which must be present in the workspace for the rule to apply. Rather, it indicates a constituent which is inserted by the Word rule into the resulting tree. Usually such dummies function as indicators of potential syntactic boundaries or serve to carry semantic information, as in the following excerpt from a clause rule in which a dummy auxiliary (D AUX) serves as a carrier for tense and mood information originally obtained through analysis of the predicate.

```
C 195  V CLS     V PRED    V NP   ...   D AUX
                 $ PS                   $ 1.3TN
                 $ NU                   $ 1.4MD
                 $ TN
                 $ MD
```

Once the algorithm has determined that a given rule has the potential for application (based upon presence of appropriate constituents), it is necessary to check more closely to see that the specified syntactic and semantic co-occurrence restrictions are met. The necessary information is found in the symbols which occur under the category symbols in the rule, each line representing a separate condition.

B. Syntactic Component

1. Syntactic Analysis

The purpose of Syntactic Analysis is to find all readings of a text span which are well-formed according to the syntactic grammar of a language. These readings are dominated by the grammar's initial symbol S. The Syntactic Analysis algorithm is now identical to the Word Analysis algorithm. It operates in one of two modes: in the first mode it builds on any reading, whether flagged or unflagged (like Word Analysis); in the second mode it only builds on unflagged readings (the original Syntactic Analysis algorithm). This algorithm operates with the syntactic grammar of a language, a grammar which has undergone considerable change during the period of algorithm development. Although originally the word component was a morphological parser and the syntactic component was a sentence syntactic parser, the design was changed to incorporate phrase level rules into the word component.

a. Restrictions and Changes - Surface Grammar

Before the algorithm applies a rule it performs basically three checks. It first determines whether the category symbols of a sequence of nodes correspond to the category symbols in the consequent of a rule. It then checks whether each node has the subscripts and values specified in the rule. Third, it performs the operations specified in the rule. Whenever these three checks are successful, it constructs the new constituent corresponding to the instructions in the antecedent of the rule.

When the algorithm compares text category symbols against rule category symbols, it constructs intermediate tables in which it records the results of the matching process obtained so far. Because of the generality of the category symbols in the word and syntactic grammars, the number of attempted rule applications and the size of the intermediate tables became so large that the maximum time allowed to the algorithm for the construction of a rule had to be increased several times. In order to reduce the size of the intermediate tables it was desirable to make the category symbols more specific, which, in some cases, caused a duplication of rules. For instance, there was the general rule which conjoined, as a single noun phrase, the sequence of a noun phrase followed by a conjunction followed by a noun phrase (NP => NP CONJ NP). This had to be separated into 1) a noun phrase followed by comma followed by noun phrase, and 2) a noun phrase followed by conjunction other than comma followed by

noun phrase. The rules most affected by this decision were the morphological rules, where adjective endings, noun endings and verb endings now had to be distinguished and the clause rules.

Fortunately, making the category symbols in clause rules more precise did not increase the number of clause rules, since this change only affected the rule antecedent. LRC established four types of clauses: CLS-MAIN, CLS-SUB, CLS-HYP, and CLS (for main clause, subordinate clause, hypothetical or interrogative clause, and 'undetermined' clause, respectively). The latter was eventually rewritten as one of the first three clauses, depending on the subscripts and values associated with it. CLS-SUB was rewritten as CLS-REL (relative clause) if its initial constituent dominated a relative pronoun.

The former rule (1)

```
V NP          V NP    V CONJ           V CLS
ˆ 2                   $ TY(COMMA)      $ TY(REL)
```

was thus changed to (2)

```
V NP          V NP    V COMMA          V CLS-REL
ˆ 2
```

With the first rule, the algorithm only determined during the subscript and value check that the sequence "the girl, he smiled" contained in "When he saw the girl, he smiled," was not a sequence of a noun phrase followed by a conjunction followed by a relative clause. With the increased amount of information contained in the category symbols of rule 2, the rule is not even tried. These changes led to a considerable reduction in analysis time. They did not, however, alleviate another problem of the analyzer. During the analysis of long sentences (consisting of more than 200 letters) which contained a relative clause, the number of the intermediate tables became so large that the time allotted for the construction of a new constituent was frequently exceeded. It was therefore necessary to sub-divide the syntactic grammar into two parts, phrase grammar and clause grammar.

Each grammar was processed during a separate analysis phase. It turned out, however, that the addition of another analysis component with its loading of programs, loading of grammars, etc., required more time than could be saved by sub-dividing the grammar. In the final solution, the scope of the word grammar was expanded to include phrasal constructions such as noun phrases

prepositional phrases, their conjuncts and expansions by post-nominal modifiers. The effect was that the size of the intermediate tables was roughly divided in half, expediting the combination of Word and Syntactic Analyses.

More than ninety percent of the current syntactic grammar used by the FORTRAN implementation consists of clause rules, rules which concatenate noun phrases and relative clauses, and rules which rewrite a relative pronoun as a noun phrase or prepositional phrase and sequences of one-word adverbs as adverbials.

2. Syntactic Choice

The primary purpose of Syntactic Choice is to disambiguate words on the basis of information contained in a sentence, and to determine the semantic function of each constituent and its semantic relations towards other constituents. In the process of making these decisions, Syntactic Choice discovers discontinuous constructions and makes them contiguous, it introduces new dictionary terms which did not occur in the text ("wuerde dies geschehen, so" becomes "falls dies geschehen wuerde, so"), it deletes words and non-terminal constituents which can be predicted (like punctuation marks) or which become features of their verb. The power of Syntactic Choice also permits the linguist to make a decision among competing (ambiguous) analyses on the basis of linguistic or pragmatic considerations.

Syntactic Choice uses information contained in the syntactic analysis rules subsequently called Choice Statement, and information which is stated in the separate Syntactic Choice Macro Grammar. It is executed in three phases. During the first phase all non-productive interpretations, i.e., those interpretations not dominated by S, are deleted. If two or more S readings dominate a text span, the S dominating the longest span is retained.

During the second phase, all Choice Statements contained in the syntactic rule and all choice rules called by the applied syntactic rules, are executed.

During the third phase, Syntactic Choice prints out the set of tentative standard sub-trees in what is called the 'parallel workspace format'. This format permits the Standard Analysis algorithms and Transfer Analysis algorithms to analyze a constituent only once, even if it occurs n times in n standard strings, i.e., if the sentence had been n times ambiguous.

As can be seen from the function of the Syntactic Choice component, the Syntactic Choice grammar is easily the most important grammar of every language dealt with by the system. The combination of choice statements associated with syntactic rules and the choice rules called by them are in effect an algorithm designed by the linguist to compute the underlying structure or structures from which the given surface structure may have been derived. The computed underlying standard structures are tentative, even more so if the sentence analyzed contains a lexical collocation. It is for this reason that a final check performed by Standard Analysis grammar is necessary.

B. Syntactic Subscript Grammar Rules

A syntactic (word or syntax) grammar rule consists of an analysis statement and, optionally, a choice statement. The former is used during Word or Syntactic Analysis; the latter, during Word or Syntactic Choice. In addition, paralinguistic information may be assigned to a rule by means of special operators: (kind of discourse, area of provenience, style, linguistic frequency, etc.), in order to permit the selection of rules for recognition and production. Additional operators of this nature will be incorporated in the rules as the necessary linguistic information is accumulated.

Consider the following rule:

```
C 263  V NP           V DET         V ADJ         V NO       D NPY
D 1    + PS(3)        ? IN          $ GD          ? GD
       + FM(DAN)      $ GD          $ CA          $ CA
       $*4.6GD        $ CA          $ NU          $ NU
       $*4.7CA        $ NU          $TTM          $TTY
       $*4.8NU        ? BF          $ DG          ? TT
       $*3.7TY        ? WD          $ IN          . 2.8,4.1
       $ 3.5DG        .*2.1,3.6     . 4.4,3.4     . 2.9,4.2
       $ 3.8EX        . 2.2,3.1     ? EX          . 2.10,4.3
       + OR(O,O2)     . 2.3,3.2
       - 2.1          . 2.4,3.3     3             7
       = 2            ? G           $ FO(X)       $ TY(X+QU)
       ^ 4            ? N
                                    5
       3              4             $ FO
       C F/1          $ FM(X+P)     $TTO

       4              5             2
       C F/1          $ CA          $ EX
                      $TTY
       5              - 2.1,3.1
       X P            - 2.2,3.2

       1              1
       A 2FM(DET)     $ FM

       6
       A 3FO(LA)

       7
       A 3TM,4TY(1TY)

       2
       X D3
       S 2-3-4-5
```

The analysis statement of a rule, such as the above, consists of the following parts: the rule consequent(s) (i.e., those constituents which appear on the right), the rule antecedent (the left-side term), optional dummy statements (here, a dummy noun phrase boundary), and a series of pre-conditions on the application of the rule and instructions to be carried out in creating the new node. (Of these parts, only antecedent and the consequent terms are used in every rule of the grammar; others are optional.) Below the analysis statement and separated from it by blank lines is the choice statement, which will be discussed in detail later in this section.

The phrase structure description (first line) of a syntactic rule is used by the analysis algorithms as an instruction to construct a new workspace term (a complex symbol) based on the workspaces previously constructed. In this rule, an NP (noun phrase) will be created from a DET (determiner), ADJ (adjective), and NO (noun). A dummy noun phrase boundary (D NPY) is inserted by the rule, rather than being part of the input description.

Two operations must be successful before a rule can be applied: term check and operation check. These checks use the information contained below the category symbols of the phrase structure description.

1. **Term Check**

The feature terms under each variable in the rule can consist of an operator, an optional modifier, a subscript name, and optional specified values.

Subscripts are represented by combinations of two letters in the column directly under the category symbol. The values of a given subscript occur in parentheses following the subscript in the rule.

The first symbol in the term is an operator which represents conditions which must be met for the rule to apply. Usually the condition indicates that a particular subscript or value must (or must not) be present. These conditions are represented by the following operators:

$ SC( ) the particular subscript SC must occur and values specified in parentheses, if any, must be present

? SC( ) the particular subscript may or may not occur with any specified values

* SC the particular subscript must not occur

Note: "* SC" may not be followed by parentheses. If a condition "subscript SC must not occur with value A" is wanted, it must be formulated as either "$ SC(*A)" or "? SC(*A)".

a. Conditions Expressed by Values of Subscripts in the Rule Consequent

1 in column S indicates that the value check is successful; 0, that it is not. 1 in column t (A, B, Z) expresses that the value t occurs in the workspace matched: 0, that value t does not occur.

(A)			(-A)			(*A)			(A^B)			(A.B)			(A,B)		
S	A	Z	S	A	Z	S	A	Z	S	A	B	S	A	B	S	A	B
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0
0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
*1)			*2)			*3)			*4)			*5)			*6)		

*1) SC must have value A,

*2) SC must have a value besides A; A is ignored;

*3) SC must not have value A; note that this condition is fulfilled if SC has no value;

*4) SC must have both the values A and B, but not as a cluster;

*5) SC must have the values A and B, as a cluster;

*6) SC must have the value A or B or both;

In addition, we use the conditions:

(Z.Z) =: must have a cluster;

(*Z.Z) =: must not have a cluster;

(-Z.Z) =: ignore cluster;

(Z) =: must have a value, cluster or not.

The operator "." between two values indicates that the subscript must have two values, for example a verb which governs two objects. Such values undergo the difference operation twice for each of the clustered values, (cf. "Operations Between Values" table which follows below).

2. Operation Check

In addition to specification of subscripts and values which must be present in the workspace for a particular constituent, the rule may also specify certain co-occurrence restrictions such as case and number agreement among constituents. This is accomplished by means of the set-theoretical operations union, intersection, and difference. Union is represented by "+", intersection by ".", and difference by "-". The usual format consists of one of the operators followed by a location statement, for example

. 4.4,3.4

which means to perform an intersection operation between the fourth subscript under term 4 and the fourth subscript of term 3. In rule C 263.1, it is between the semantic type of the noun and the required type of the modificand of the adjective. (The T between the operator and the subscript in $TTY and $TTM means that the value tree must be consulted for this subscript in performing any operations.)

The three set-theoretical operations may be summarized as follows:

Operations

. 2.1,3.3	+ 2.1,3.3	- 2.1,3.3
find common values of WSC 2.1 and WSC 3.3	add values of WSC 2.1 and WSC 3.3	subtract common values of WSC 2.1 and WSC 3.3 from values in WSC 3.3

We use "WSC n.m" to refer to the workspace subscript which corresponds to the location specified by n.m.

A complete chart of the results of the operations between various combinations of values follows.

The operations to be performed during the operation check are stated in each rule subscript statement, which has the general form: operator, modifier, and the location statement n.m, k.j. This statement indicates that the operation is to be performed with the values of the mth rule subscript in the nth rule term and the values of the jth subscript in the kth rule term.

The operations can be modified by statements in the modifier column: no modifier indicates that the values of the matched workspace subscript are to be used; operations can further be modified as to: must be successful, must not be successful, need not be successful, which are represented by blank, *, ?.

Sometimes a T is used between the operator and subscript, for example $TTY. This T indicates that a value tree is to be consulted in carrying out the operation regarding subscript TY. (T replaces redundancy rules since such rules introduce a large number of values during analysis, the greater part of which are not actually needed for the interpretation of the sentence analyzed). The nodes of such trees are values. A value P can be subsumed under a value Q if an upward path leads from P to Q. Thus, the value trees are nested sets of values such as those for the semantic classes for nouns. Cf. the following excerpt from the value tree for noun semantic type:

```
                 PO
          (physical object)
                 /\
                /  \
               /    \
              /      \
            AN        IN
       (animate)    (inanimate)
            /\
           /  \
          /    \
         /      \
       HU        AL
    (human)   (animal)
```

Different value trees may exist for different category/subscript symbol combinations; the same tree may be shared by different category/subscript symbol combinations.

If any of the operations specified is not executed as stated, the rule is rejected. The operations Intersection and Difference are successful whenever the subscripts involved have a value in common. A value preceded by "-" is excluded from the specified operations. The operations referring to a subscript with the operator "?" are ignored if that rule subscript has no match in the workspace.

After the operation check is successfully performed, the new workspace is constructed.

Operations Between Values

Intersection . (period)	Second Argument				
First Argument	A	B	A+C	B+C	A+D
A	A	0	A+C	0	A+D
B	0	B	0	B+C	0
A+C	A+C	0	A+C	0	A+C+D
B+C	0	B+C	0	B+C	0
A+D	A+D	0	A+C+D	0	A+D

Intermediate Intersection -. (minus-period)	Second Argument				
First Argument	A	B	A+C	B+C	A+D
A	A	0	A	0	A
B	0	B	0	B	0
A+C	0	0	A+C	0	0
B+C	0	0	0	B+C	0
A+D	0	0	0	0	A+D

Difference - (minus)	Second Argument				
First Argument	A	B	A+C	B+C	A+D
A	NIL	0	NIL	0	NIL
B	0	NIL	0	NIL	0
A+C	0	0	NIL	0	0
B+C	0	0	0	NIL	0
A+D	0	0	0	0	NIL

Summation + (plus)	Second Argument				
First Argument	A	B	A+C	B+C	A+D
A	A	B,A	A+C	B+C,A	A+D
B	A,B	B	A+C,B	B+C	A+D,B
A+C	A+C	B,A+C	A+C	B+C,A+C	A+D,A+C
B+C	A,B+C	B+C	A+C,B+C	B+C	A+D,B+C
A+D	A+D	B,A+D	A+C,A+D	B+C,A+D	A+D

Note: 1. no upward path leads from A to B, or B to A;
2. -. = intermediate intersection (for subsequent difference operation);
3. 0 = not successful,
NIL = successful, the result is the empty set.

3. Workspace Construction

The new workspace antecedent is constructed by means of the instructions stated in the syntactic rule antecedent, based on the information found in the syntactic rule and in the workspaces which were matched by the terms in the rule consequent.

The new subscripts are created either by "carrying" (copying or changing) the "old" subscripts (of the workspaces matched by the rule terms) or by defining new ones. The new subscripts are assigned values in the subscript statement of the syntactic rule antecedent.

a. Carry Operations

Five carry operations, identified by the operators "$", ".", "-", "=", and "^", can be executed.

1] $ Carry the workspace subscript.

a) $ n.m Copy (with its values) the mth subscript matched in the nth rule term (subscript n.m).

b) $*n.m SC Assign as values to subscript SC the result of the operation in the mth subscript of rule term n.

c) $.n.m SC Assign as values to subscripts SC the result of the intermediate intersection of the difference operation stated in the mth subscript of rule term n.

We shall, subsequently, represent any of these three operations by the expression $_n.m.

2] . Change the values of the workspace subscript.

a) .n.m(A) Copy the subscript matched by the rule term n.m and change its values to A.

b) ._n.m SC(A+X) Perform operation $_n.m and add A to the values of the new subscript.

c) ._n.m SC(-A) Perform operation $_n.m and delete, if it occurs, the values A from the values of the new subscript.

3] = n	Copy all subscripts matched by those in the rule n which are not referred to (by "n.m") in the rule antecedent.

4] - n.m	Do not copy the workspace subscript matched by the rule subscript specified in n.m into the new workspace. (This statement only occurs in combination with = from 3] above. "Do not carry a workspace subscript" is normally performed by not mentioning the matching rule subscript in the rule antecedent.)

5] ^ n	Copy all those workspace subscripts which have not been mentioned specifically in rule term n.

The ^ operator is extremely useful, since by its use it is possible to write rules without losing subscripts that will be relevant for other rules; e.g., the rule,

```
   (1)              (2)              (3)

V  VB            V  V             V  EN
+  PS(1,3)       +  CL(40,41      B
+  NU(P)            ,43)          L
+  TN(PA)
+  MD(I,S)
+  FM(PAPL)
^  2
```

constructs the higher new workspace (VB) from the lower old workspaces (V) and (EN), below.

```
              (1)

              VB
              TS(HU)
              TO(PO)
              PS(1,3)
              NU(P)
              TN(PA)
              MD(I,S)
              FM(PAPL)
               /\
              /  \
             /    \
            /      \
           /        \
        (2)          (3)

        V            EN
        CL(41)
        TS(HU)
        TO(PO)
```

(The condition check is successful since both rule terms 2 and 3 are sub-configurations of the corresponding workspaces 2 and 3. Statement ^ 2 specifies that all subscripts and values of workspace 2 which are not matched by a subscript in rule term 2 are to be added to the new workspace.)

b. Dummy Terms

The terms which occur in the dummy statement of a grammar rule are terminals of the standard (deep structure) grammar of that language, even though they are not surface structure constituents. They are constructed in the same way as new workspace terms. Dummy terms are lexicalizations of the grammatical information obtainable from the subscripts and values of a constituent and/or from its position in the rule consequent. Thus they may be used to carry information such as tense, which may be carried on the surface by either the verb form or by an adverb such as German "morgen" or English "tomorrow". A dummy may carry information which is marked by word order, for example the distinction between statements, imperatives, and questions. Dummy terms constructed for a rule are stored in the list associated with this particular application of the rule.

c. Cover Symbols

The letters X, Y, and Z, each optionally followed by a digit n, may be used as variable names for value symbols. Repeated occurrences of the same combination of Xn in a rule represent the same symbol. Cover symbols used in a rule antecedent but not in the rule consequent represent the contents of the subscript in the consequent referred to in the antecedent. Since the cover symbols do not contribute to the exclusion of a grammar rule during the term check, they have been used in only a few rules.

d. Clustering of Subscripts and Values

The connector "/", following a subscript statement in a word or syntactic grammar rule, links the ith value set of a subscript with the ith value set of the following subscript. More than two subscripts can be connected by means of "/". Value sets are separated by ' (apostrophe), as in the dictionary level rules. (See the section on conflation of dictionary rules for additional explanation of this convention.)

4. Choice Statement

Choice operations only apply if Syntactic Analysis has been successful and then only to those constituents which participate in the successful analysis. The Choice statement sets specified in the Choice portion of a rule are tests and transformations. Since these transformations are associated with the applications of a rule, it follows that their structural description is (in general) met if the rule has been applied. These transformations are subscript/value-sensitive; thus additional conditions can be imposed on their application. Conditions and associated operations are linked by means of identical numerical subscripts. Some choice sets may be associated with an optional name.

a. Choice Conditions

The conditions that can be specified within a choice set are:

term check
operation check
(previous) transformations check

Term check establishes whether one or more constituents have (do not have) specified features. Operation check establishes whether certain relations hold between the features of stated constituents. Transformations check determines whether a previous transformation or combination of transformations has been (has not been) successfully applied. Term and operation check correspond to the "structural description" of a transformation; transformations check, to "trans-derivational constraints". It is further possible to add an "if true go-to" and "if false go-to" instruction to a condition which directs the choice algorithm to a particular subsequent transformation, dependent on whether the condition was satisfied or not.

b. Choice Operations

The choice operations that can be performed are:

Assignment of values, which disambiguates ambiguous lexical items in disambiguating environment (within a sentence);

Deletion of constituents;

Superscript assignment, which permutes constituents;

Call choice-rule macro;

Construction and insertion of additional dummy terms;

Rejection of rule.

1) Assignment of Values

The assignment statement has the format A-Modifier-Term-SC-(Term-SC)-Value; for example:

V	CLAUSE	V	SUBJECT	V	OBJECT	V	PREDICATE
	$.3.2FO	$	NU	$	FO	$	NU
		$	PS	-	3.1,4.3	$	PS
		.	2.1,4.1			$	FO
		.	2.2,4.2				

1
A 3FO,4FO(1FO)

The assignment statement is read: Assign the values of the subscript FO in term 1 to the subscripts FO in terms 3 and 4 if either is ambiguous. The ambiguity can be recognized from the occurrence of a "trace" linked with the subscript FO in term 1.

In the address provided by the "trace" the assignment operation weights (gives preference to) the value(s) contained in 1FO and the subscripts which contain the(se) value(s). If necessary, the "value-tree" is consulted.

2) Deletion of Constituents

Constituents can be deleted by two requests:

a) X Dn

b) S n(0)

where n refers to the nth rule term. The first request deletes a constituent C which is immediately dominated by the rule antecedent; the second request deletes constituent C and every constituent dominated by C.

3) Superscript Assignment

Superscript instructions associate a standard order (Normal Form order) with the constituents of a rule after surface analysis (Standard Analysis), based on the properties of the constituents interpreted by the rule, and, if specified, on the relations between those constituents.

There are two types of superscript instructions: name-assigning instructions and order-defining instructions. The first type has the format S n(T), where T is an alphanumeric symbol, or the form P:m, or F:m for "precedes term m", or "follows term m". The order-defining instruction has the form S N1-N2-...-Nm, where each Ni is an assigned name. The order of the Ni's in the instruction represents the order of the constituents in the standard string.

4) Call Choice Rule

This request has the form R NAME(n1,n2,...nm), where NAME is the name of the choice rule and the ni's are the rule constituents used by the choice rule.

Choice rules have the format of syntactic rules. Their analysis part consists of category symbols only, they may contain dummy statements (cf. e in "Choice Operations" above) and must contain a choice statement, normally with a large number of choice sets.

Choice rules are mainly used as an abbreviatory instruction. They reduce the amount of information necessary to be stored with a syntactic rule, as well as reducing the time for actually writing a syntactic rule. Thus, the twenty-two choice sets associated with the choice rule CT, which determines the "type of clause", can be conflated to the request R CT(n).

5) Rejection of Syntactic Rule

This request is represented by the statement C T/E or C F/E, which stands for "if the stated condition is true (false), reject the rule". This instruction will be executed whenever a string meets the syntactic requirements of a rule but violates the semantic requirements. This instruction also permits the selection of one analysis in favor of another in case of multiple interpretations of a string.

5. Strategy for Clause Description

German clause level constituents can be permuted to a large extent. The number of clause patterns made possible by the rearrangement of constituents is not affected by the subscript grammar. In order to reduce the number of clause patterns, it was necessary to restrict the categories which were permitted to occur on clause level. Those which have been utilized are: surface subject, predicate, surface object, and adverbial.

a. Surface Subject

The surface subject appears as,

```
V  NP
$  CA(N,CL)
```

which signifies that the surface subject is either a noun phrase in the nominative case, or a clause. The surface subject may dominate noun phrases and subject clauses; among the latter are "dass"-clauses and verbal clauses. The German word "es" is interpreted both as an adverbial and as a noun phrase. In its anticipatory usage it is interpreted as adverbial as in,

> "Es befanden sich drei Leute im Zimmer.
> Er hatte es aufgegeben, ihn davon abzubringen."

b. Predicate

The predicate is realized in three versions:

```
V  PRED
V  PRED  ...  V  PRFX
V  MODAUX  ...  V  VERBAL
```

PRED dominates the finite verb form of a regular verb or a modal or auxiliary which is used as a full verb; it also dominates the concatenation MODAUX-VERBAL and VERBAL-MODAUX.

PRFX dominates separable prefixes.

MODAUX dominates finite forms of modals and auxiliary verb forms; the German verb "lassen" is classified as being a possible modal, as in

"Er liess den Mann von dem Detektiv beobachten."

The German verbs "bekommen" and "erhalten" are classified as potential auxiliaries which form the passive, as in

"Er bekam (erhielt) ein Buch geschenkt."

VERBAL dominates non-finite forms of full verbs, "zu"-infinitives, and concatenations of non-finite verb forms and non-finite modals or auxiliaries.

c. Objects

Objects are realized as

V NP
$ CA

This symbol dominates noun phrases and object clauses (cf. "Strategy for Clause Description" above).

d. Adverbials

Adverbials appear as

V ADV
? CA

They dominate one-word adverbs, prepositional phrases, prepositional objects, subordinate clauses, and noun phrases which function as adverbials of extension in space and time, as "vierzehn Jahre" in

"Er arbeitete vierzehn Jahre."

6. Information in Clause Rules

The decisions that need to be made in the syntactic part of a clause rule may be described by discussing the following rule.

```
FORM SYNTACTIC

          (1)         (2)          (3)          (4)          (5)

C 186   V CLS       V NP         V NP         V ADV        V PRED
        $ 5.1FL     $ CA(N,CL)   ? POS(MED)   * ES         $ FL
        $ 5.2PX     ? POS(IN)    $ CA         ? POS(MED)   ? PX(LA)
        $ 5.5TN     ? FM         - 3.2,5.8    ? CA         $ PS
        $ 5.6MD     $ PS                      -?4.3,3.3    $ NU
        $*2.6       $ NU                                   $ TN
        $.3.3FO1    . 2.4,5.3                              $ MD
        $ 2.8WD     . 2.5,5.4                              ? AX
        $ 2.9G      ? WD                                   $ FO
        $ 2.10N     ? G                                    F
        $ 2.3FM     ? N
        $*3.3FO     P
        $ 5.7AX
        $.4.4A1

        1
        R CT(5)

        2
        R SPECAV-1(4)

        3
        C 1.20,F/5

        4
        C T/11,F/E
        R PC:V2OC(5,2,3,4)
        S 10(VC)

        5
        C 1.21,F/7

        6
        C T/11,F/E
        R AC:VSOC(5,2,3,4)
        S 9(VC)
```

C 186 (Cont.)

(1)

7
C 1.22,F/9

8
C T/11,F/E
R CC:SOC(2,3,4)
S 5(P)
S 9(VC)

9
C 1.5!1.8,F/E

10
C F/E
R LC:VSOC(5,2,3,4)

11
R EX-3(2,3,4)

12
X D5
S 6(L)
S 7(R)
S 8(F)

13
S L-S-N-
SP-VC-F-
M-P-LC-
O-O2-1-
2-3-4-R

C 186 (Columns 6-10)

(6)	(7)	(8)	(9)	(10)
D LEFT	D RIGHT	D AUX $ 1.3TN $ 1.4MD	D VOICE + VC(A)	D VOICE + VC(P)

The consequent part of rule C 186 (terms 2-5) analyzes any string consisting of a surface subject, followed by a surface object, followed by an adverb, followed by a predicate. The subject must either be in the nominative case or it must be a clause. The subject must also agree in person and number with the predicate.

The predicate may be composed of an auxiliary and a non-finite verb form or it may be a finite verb. The predicate must not occur with the prefix (i.e., must have the feature [PX(LA)]), and it must govern the case of the surface object and may govern the case of the surface "adverb" if it is a prepositional object.

The antecedent of rule C 186 stores the tense and mood information of the predicate as well as information about the type of auxiliary which occurs if the predicate consists of a finite verb and a non-finite verb. The antecedent also stores the information that the predicate is in final position [FL]. It stores the case of the governed object [FO1] and the case information, if any, of other objects which the verb governs obligatorily [FO]. It stores the preposition of the "adverb" if it is a prepositional object [A1].

If the subject is a pronoun, the antecedent stores information as to whether it is an interrogative pronoun [FM], or a relative pronoun [WD,G,N].

a. Choice Rules

1) Function

It is the function of the choice rules associated with a particular syntactic rule:

to determine the deep structure of the string interpreted by the rule, based on the semo-syntactic features associated with the clause constituents, and

to generate that deep structure, called standard string.

This is performed by permuting the clause level constituents, by adding new terminal symbols (dummy terms or standard terminals), and by deleting certain surface terminals, such as prefixes or the reflexive pronoun in cases of actual reflexive verbs. ("sich waschen" vs. "sich beeilen"; "ich wasche ihn", but not "ich beeile ihn.")

An additional function of the choice rules is the elimination of forced ambiguous readings.

2) Determination of the Deep Structure

The deep structure of a given sentence is determined in several phases.

a) Determination of Clause Type

We distinguish between active clauses, passive clauses, copula clauses, and "lassen" clauses. "Lassen" clauses are further divided into "lassen" clauses with an embedded active clause, or with an embedded passive clause. Example:

(Active) "Er liess den Jungen den Hund schlagen."

(Passive) "Er liess den Hund von dem Jungen schlagen."

The clause type is determined by evaluating the result of the intersection between the values of the subscript TY of the constituent MODAUX and the values of the subscript AX of the non-finite verb part. The actual decisions are made by choice rule CT which is called with the information contained in the rule antecedent and the full verb (PRED or VERBAL).

PRODUCTION SYNTACTIC

```
 C  1          V  CT              V  CT

               1                  1
               C  F/7             $  CT(-M)

               2                  2
               C  F/5             $  CT(MI)

               3                  3
               C  T/21            $  CT(M)
               X  NHABEN+ZU
                                  5
                                  $  CT(L)

               4
               C  *3,T/20         9
               X  NSEIN+ZU        $  VC(P)

               5                  13
               C  T/22            $  ZU
               X  NLASSEN
                                  16
               6                  *  VC(A)
               C  *5,T/20
               X  NBEKOMMEN       22
                                  $  TY(C)

               7
               C  F/12
               $  AX(M)

               8
               C  T/22
               $  AX(M+L)
               X  NLASSEN

               9
               C  T/20
               X  NMODAL+PASSIVE

               10
               C  T/R
               $  AX(A)
               X  NWERDEN+
                  PASSIVE-OR-
                  MODAL+ACTIVE
```

C 1 (Cont.)

```
11
C   *10,T/21
X   NMODAL+ACTIVE

12
C   F/10
X   AX(HZ!SZ!A!H!A+S)

13
C   F/16
$   AX(HZ!SZ)

14
C   T/21
$   AX(HZ)
X   NHABEN+ZU

15
C   *14,T/20
X   NSEIN+ZU

16
C   T/20
$   AX(A+S)
X   NSEIN+PASSIVE

17
C   *16,T/21
X   NPERFECT-ACTIVE

18
C   T/20
$   AX(A+W+B)
X   NBEKOMMEN

19
C   *18,T/20
X   NWERDEN+PASSIVE

20
C   4!6!9!15!16!18!19,
    T/R
X   NPASSIVE
```

C 1 (Cont.)

21
C 3!11!14!17
X NACTIVE

22
C 5!8!11!17,T/R,F/R
X NCOPULA

Note: The decisions made in this rule are represented by the graphs on the following pages.

```
                       Type of Clause
                       --------------

                              1     expanded VERBAL?
                             / \
                            /   \
                         + /     \ -
                          /       \
                         /         \
      contains "zu"?    2           7
                       / \       --------
                      /   \      | see  |
                   + /     \ -   | next |
                    /       \    | page |
                   /         \   --------
   contains       3           5  contains
   "haben"       / \         / \  "lassen"
              + /   \ -   + /   \ -
               /     \     /     \
              /       4   /       6
       HABEN+ZU   SEIN+ZU  LASSEN   BEKOMMEN
           |           |    |        |
           |           |    |        |
         [21]       [20]  [22]     [20]
           |           |    |        |
           |           |    |        |
        ACTIVE    PASSIVE  COPULA   PASSIVE
           |           |    |        |
           |           |    |        |
         [22]         [R]  [R]      [R]
           |
           |
          [R]
```

The rule numbers in the graph correspond to the numbers in the choice rule.

Type of Clause (continued)

```
                                  7    modal?
                                ---------
                                | from  |
                                | prior |
                                | page  |
                                ---------
                                  /  \
                                 /    \
                                /      12
                               /     --------
                              /      | see  |
                             /       | next |
                            /        | page |
                           /         --------
                          8   Lassen?
                         / \
                      + /   \  -
                       /     \
                      /       9   Passive?
                     /       / \
                    /     + /   \  -
                   /       /     \
                  /       /       10   also Aux?
                 /       /       /  \
                /       /     + /    \  -
               /       /       /      \
          LASSEN   MODAL+    WERDEN+    MODAL+
            |      PASSIVE   PASSIVE    ACTIVE
            |        |         OR         |
            |        |       MODAL+       |
            |        |       ACTIVE       |
            |        |         |          |
           [R]     [20]       [R]       [21]
                     |                    |
                  PASSIVE              ACTIVE
                     |                    |
                    [R]                 [22]
                                          |
                                       COPULA?
```

Type of Clause (continued)

```
                          12    Haben and/or Sein?
                        ---------
                        | from  |
                        | prior |
                        | page  |
                        ---------
                            /\
                           /  \
                     +    /    \    -
                         /      \
                        /        \
                       /          \
                      /            \
               13  Zu-inf.?         \
                  / \                \
               + /   \ -              \
                /     \                \
           14 Haben?  16 Passive?       18  Bekommen?
             / \       / \                 / \
            /   \     /   \               /   \
      HABEN+  SEIN+  SEIN+    PERFECT  BEKOMMEN  WERDEN+
      ZU      ZU    PASSIVE   -ACTIVE     |      PASSIVE
       |      |       |          |        |        |
       |      |       |          |        |        |
       |      |       |          |        |        |
     [21]   [20]    [20]       [21]     [20]     [20]
       |      |       |          |        |        |
       |      |       |          |        |        |
       |      \       /          |        \        /
     ACTIVE    PASSIVE        ACTIVE        PASSIVE
       |          |              |             |
       |          |              |             |
     [21]       [20]           [21]          [20]
       |          |              |             |
       |          |              |             |
     COPULA?     [R]          COPULA?         [R]
```

The decisions made by this choice rule are not final; the actual determination of the CLAUSE TYPE is dependent on the relations between the predicate and its complements. Thus, "addieren" ("to add"), will first be interpreted as active voice in a sentence such as "die Zahlen addieren sich zu hundert." The verb complement choice rule AC:VSOA (for active complete: verb, subject, object, adverb) will, however, assign to this clause the interpretation "passive voice". This permits both the translation "the numbers are added up to a hundred" and "the numbers add up to a hundred". Similarly, "es wird getanzt", which superficially looks like a passive sentence, will be interpreted as an active sentence with a deleted agent, which permits the translations "they danced" or "people danced".

b) Determination of Adverbials

The choice rule called SPECAV (special adverb) determines whether an adverb is the negation "nicht" ("not"), or an adverb of the type "gern", "lieber", "weiter" which function as deep predicates. The negation is moved directly behind the subject to facilitate the generation of the English output; it could, however, be put in front of or behind the actual clause to indicate its operator status. This difference in treatment would not have any effect on the translation. The special adverbs of the type "gern", "lieber" are moved into the predicate position. The surface predicate and its object complements are treated as the clause complement of the predicate represented by the surface adverb.

Adverbials which dominate prepositional phrases undergo further checking in the verb complement choice rules. There, we determine whether or not the adverb is a prepositional object of the predicate. In the final assignment statement, each adverb is assigned a numerical value (cf. Choice set number 12 in rule C 186 above). These numerical values are deleted if they compete with an alphanumeric name such as N, SP, or O, O2, O3.

c) Verb Complement Rules

We distinguish four types of verb complement rules: those which contain a passive predicate, an active predicate, a copula, or a form of "lassen". The corresponding choice rules begin with the letters PC, AC, CC, and LC. For each set of these choice rules there exists as many alternates as there are basic sentence patterns. The functions of verb complement rules are basically four-fold:

a. they determine whether the subject and object of the verbs agree with the verb in syntactic surface appearance and semantic type;

b. if these tests fail, a test for the occurrence of different clause type (remember "addieren sich" and "wird getanzt" above) is made;

c. after that there is a test for the occurrence of a lexical collocation. This test is executed only superficially, by checking whether the verb and the constituent in question agree in their values of LC;

d. if all of these tests fail, the main rule which called the choice rule is rejected.

d) Superscript Assignment

The clause constituents are finally connected in the sequence represented by the final superscript assignment (statement number 13 in rule C 186 above). The sequence L-S-N-SP-VC-F-M-P-LC-O-O2-O3-1-2-3-4-R stands for "left boundary, deep subject, negation, special adverb, voice information, tense information, auxiliary or modal, predicate, lexical collocation, first deep object, second deep object, third deep object, first adverb, second adverb, third adverb, fourth adverb, right boundary". Of these, only left boundary, subject, voice, tense, predicate, and right boundary are obligatory. If a constituent is assigned more than one alphabetic superscript name, the Cartesian product of permissible standard strings will be generated, with the provision,

a. that no two identical names may occur in the same standard string, and

b. that identical standard sequences which were derived by means of assigning different names to the same constituent are conflated to one standard string.

Currently, choice rules can only be called from a main rule, but we plan to extend the algorithms' capabilities to that of calling a choice rule from a choice rule. However, a restriction will be imposed, in that a choice rule which had been called by another choice rule may not be call a third choice rule.

IV. GERMAN STANDARD (DEEP) STRUCTURE

A. Economy of Standard Description

The Standard Grammar [GSG] is essentially a deep structure grammar. The number of Standard Grammar rules is fairly small. There are three reasons for this:

the constituents occur in their deep structure order,
relevant boundary information is retained by means of dummy terminals, and
surface structure which is identical to deep structure is retained.

From the structures which remain after the permutations and deletions associated with Syntactic Choice, the Standard Grammar builds deep structure sentences which have normalized constituent order. If no constituents have been permuted or deleted, the Standard tree is very similar to the surface tree. It differs only in the addition of a few dummy nodes which are realized as terminals in the Standard Grammar and which carry information such as tense and mood. Extensive application of Standard rules is only necessary where Syntactic Choice has destroyed grammatical structure.

Surface structure is destroyed in those cases where each node labeled clause, or dominating a clause, or dominating a standard expanded adjective, is destroyed by a transformation in Syntactic Choice. Adjectives are expanded if they concatenate with an object complement or a sentential adverbial. Strings containing an expanded adjective phrase are rearranged to represent Standard order.

Clause rules are destroyed because each clause rule introduces at least three dummy terms and these dummy terms must be incorporated into the Standard structural description. (The obligatory dummies are D LEFT, D RIGHT, and D AUX representing left sentence boundary, right sentence boundary, and auxiliary, respectively.)

B. Standard Clause Patterns

Since word order is not a consideration in deep structure rules, all sentences having been permuted to the order Subject - Verb - Objects - Adverbs by Syntactic Choice, there are only a few basic patterns in the Standard clause rules. These are structures consisting of--

subject, predicate, and no object
subject, predicate, and exactly one object
subject, predicate, and two objects
subject, predicate, and three objects.

Each of these patterns may occur with one or more adverbials following the nominal elements and separated from them by a verb phrase boundary (introduced as D VBY in Syntactic Choice).

In the Standard grammar, all nominal elements such as subjects, objects, etc., are analyzed as ARG (arguments). All sentential modifiers such as subordinate clauses or prepositional phrases are realized as ADV (adverbials).

Consider this clause rule from the German Standard Grammar.

```
                (1)          (2)       (3)          (4)          (5)

C 40206    V CLS          V LEFT    V ARG        V PRED       V ARG
D 2        $*5.8LX                  $TTY         $ IO(O.*A)   $ OR(O)
           $*3.2TS                  . 3.1,4.4    $TTO         $TTY
           $.5.4TO                  ? LX         $TFO         $TCA
           $.5.5FO                  +?3.3,4.5    $TTS         - 5.2,4.2
                                                 ? LX         -?5.3,4.3
                                                 +?3.4,5.7    - 5.1,4.1
           1                                                  ? LX
           A 3TY,4TS(1TS)                                     +?4.6,7.1
           A 4TO,5TY(1TO)
           A 4FO,5CA(1FO)
           A 4IO(O)

           2
           $ LX(P)
           X P

   C 40206.2 columns 6 to 8 >  (6)          (7)          (8)

                               V VBY        V ADV        V RIGHT
                                            ? LX
```

The symbol VBY stands for verb phrase boundary. It is introduced by the verb complementation choice rules.

As can be seen from rule C 40206.2 above, Standard Analysis is followed by Standard Choice. There are only two types of instructions executed in Standard Choice: assignment statements, which help to select the proper translation equivalents, and superscript assignment statements, which change the order of the standard terms to the universal order if the two should be different.

V. LEXICAL COLLOCATIONS (Idiomatic Expressions)

Although the syntactic rules of the German Standard Grammar are very similar to (though considerably less complex than) their counterparts in the German Syntactic Grammar and so do not require further comment, there are also, somewhat unexpectedly, standard word rules. These are necessitated by special circumstances, namely certain idioms and quasi-idiomatic expressions.

Idiomatic expressions which contain no internal variables and whose elements always occur in a set order with no other elements intervening can be handled at the level of the German Surface Grammar simply by writing a lexical rule which contains blanks, as is done for certain set phrases such as these which follow:

```
C 6259          V ADV     * VOR $CHRISTUS
T GER DICT      P         P
                \         F

C 6270          V ADV     * FUER DIE $PRAXIS
T GER DICT      P         P
                \         F
```

However, most idioms cannot be handled in such a straightforward manner. First, some idioms contain other elements which may vary from one sentence to another. Also, some idioms do not have a fixed word order, and so would need more than one dictionary entry. For this reason, most idioms are analyzed only after the string has been permuted into a standardized word order, namely after Syntactic Choice. This is accomplished by means of the rules of the German Standard Word Grammar and a special part of the Standard Analysis program. It is referred to as Lexical Collocation, since many of the expressions treated in this manner contain discontinuous constituents located in more than one continuous span of the sentence string.

It is the purpose of Lexical Collocation analysis to find all sequences of terminal symbols which form what we call a lexical collocation, without affecting the structural interpretations of the individual component, since these might be needed to determine possible transformations.

In a surface text, elements of a lexical collocation can occur in any order and at any distance, that is, they may occur discontinuously. After Syntactic Choice, however, we can determine precisely, if a lexical collocation occurs, which order the individual elements will occur in, their syntactic superstructure, and which constituents, if any, occur between them, obligatorily and optionally. Consider the rule,

```
V V        C 145     V NP     V ADV     C 100.8     C 23000
+                             O 1                   V ADV
+                                                   D 5,6
.
.
.
R 2,6
S 2-6-3-4
```

which interprets "take into account". This rule is to be read as: The sequence of terminals C 100.8 ("into") and C 23000 ("account"), which is dominated by the constituent ADV covering the text span from the 5th to the 6th rule term, may form a lexical collocation with the dictionary entry C 145 (take, took, taken). Between these terminals, adverbials may optionally occur (signified by the operator O below V ADV). The maximum number of adverbials which may occur in this position is 1 (O 1). A noun phrase must occur between the terminal "take" and the sequence "into account". The rule antecedent V V (verb) rewrites only the second and sixth rule term; the rule terms in the rule consequent are to be arranged in the order 2, 6, 3, 4 (the rule antecedent is always counted as rule term 1). The application of the rule above to the structure,

```
                                     ADV
                                      |
                                     PRPH
                                     /|
                                    / |
                                   /  |
                                  /  NP
                                 /    |
             NP                 /    NO
             |                 /      |
    V       PRN            PREP       N
    |       / \             / \      / \
    |      /   \           /   \    /   \
    |     /     \         /     \  /     \
   took   it               into    account
```

would thus result in the structure

```
                         V
                        /|
                       / |
                      /  |
                     /  ADV
                    /    |
                   /     |
                  /     PRPH
                 /      /|
                /      / |
               /      /  |
              /      /   NP
             /      /    |
            /      /     |
           /      /      NO      NP
          /      /       |       |
         /      /        |       |
        V     PREP       N      PRN
       /|      /|       /|      /|
      / |     / |      / |     / |
     /  |    /  |     /  |    /  |
    took    into    account   it
```

A. Entries Without Internal Variables

'Verbal lexical collocations' are set phrases consisting of a verb and either a noun phrase, a prepositional phrase, a non-finite verb form, an adjective, or an adverb, e.g., "erfolgen" = "take place"; "ins Gewicht fallen" = "be important". Since such lexical collocations have features and meanings which may not be derived from their individual components, they must be treated as lexical phrases. However, since their components frequently occur discontinuously and in various sequences, they must be handled differently from normal verbs. For this purpose the subscript LC (lexical collocation) was added to the dictionary entries of those nouns and verbs which may occur as components of lexical collocations. Surface analysis refers to those subscripts and guarantees that the items occur contiguously and in a pre-defined order in the so-called "standard string", which is generated after surface analysis.

For the actual analysis and translation of lexical collocations, standard dictionary rules were coded which are applied to the standard strings.

The constituents of standard strings are the dictionary readings of the underlying lexical elements. Standard dictionary rules concatenate these readings in multi-branch rules and assign to the whole structure the syntactic and semo-syntactic features described for the general verb system earlier in this report. Thus, the German standard dictionary rule C 5011 (which appears below) analyzes the lexical collocation "ins Gewicht fallen" ("be important"):

```
                        C  5011

                        V
                        FS(N)
                        TS(E)
                        IS(S)
                        FO(LA)
                        TO(LA)
                        IO(LA)
                          |
       -----------------------------------------
       |           |            |             |
       |           |            |             |
     C 4201      C 9028       C 149       C 20868

      fall         in           s          Gewicht
```

To this standard rule (C 5011) the following German normal form (transfer) rule applies:

```
V BE+IMPOR    C 5011    C20868    C 149     C 9028    C 4201
  TANT                            A 1       A 1       A 1
A CAT(V+P)                        B 2       B 3       B 4
N TM(INS+G
  EWICHT+F
  ALLEN)
```

The corresponding English normal form rule is in the same translation equivalence class, BE+IMPORTANT:

```
V BE+IMPORTANT    C 5123    C 20    C 10377
                                    A 1
A CAT(V+P)                          B 3
```

The right-hand side of this rule refers to the English standard dictionary rule C 5123, which, in turn, generates the English standard string:

```
                    C 5123

                    V
                    FS(N)
                    TS(E)
                    IS(S)
                    FO(LA)
                    TO(LA)
                    IO(LA)
                   /      \
        C 20      /        \   C 10377
                 /          \
              BE        IMPORTANT
```

(Note: transfer rules are described in more detail in the next section. Here, however, one should note that the phrase is translated as though it were a single lexical item.)

The correct endings and morphological variants (in this case "am", "are", "is", etc.) are generated by the English re-arrangement grammar.

German verb phrases which we call 'hidden passive phrases' -- i.e., those which contain empty function verbs such as "gelangen zu" and "kommen zu", followed by nominalized verbs -- are also treated as lexical collocations. Examples are:

"zur Ausstossung gelangen" = "be ejected"
"zum Einsatz kommen" = "be employed"

An additional subscript P identifies these German phrases as passive in meaning to guarantee their correct translation. The English translation equivalents need not be coded as phrases.

B. Lexical Collocations with Internal Variables

Some lexical collocations contain variable internal slots, as for example the noun modifier slot in "to take V care that...", (where V again stands for "variable"): "he took care that...", "he took GREAT care that...", "he took THE GREATEST POSSIBLE care that...", etc. For such phrases, standard rules are written which provide for variables in their right-hand sides (cf. rule C 5120 below). Since the present rule format does not

allow optional rule constituents, several rules were coded for such phrases, one for each possible string. As an example, the English standard rules for "take care that" are shown here:

```
C 5119  V V          C 4726       C 23195
          + FS(N)    $ PX(LA)
          + TS(HU)
          + IS(S)
          + FO(CL)
          + TO(TH)
          + IO(O)

C 5120  V V          C 4726       V ADJ   C 23195
          + FS(N)    $ PX(LA)
          + IS(S)
          + FO(CL)
          + TO(TH)
          + IO(O)
```

where C 4726 is the rule number for "take" and its allomorphs and C 23195 the number for "care".

For some phrases, additional rules may be necessary to allow for optional determiners, noun modifiers, and plural noun endings, e.g., "to pose (DET)(ADJ) problem(s)". These lexical collocations do not constitute set phrases, but rather, instances in which a verb has a specific and unusual meaning (and translation) in the environment of a noun phrase whose head noun is a particular lexical item. Beyond this, all normal rules of NP analysis and generation apply.

VI. NORMAL FORM GRAMMAR

The context-free rules of the normal form subscript grammar, subsequently referred to as Normal Form grammar or NF grammar, differ from surface and standard rules in two respects: they apply to connected graphs; they are not rewrite rules.

An NF rule applies to all graphs -- terminal, non-terminal, or combinations of them -- whose nodes, labeled by complex symbols, are non-distinct from the complex symbols in the consequent of the NF rule. The antecedent of the NF rule assigns to all graphs to which it applies a particular semantic reading, an NF expression, represented by that antecedent. Since NF expressions apply to those graphs whose nodes are labeled by complex symbols, it is possible to assign a particular NF reading to a terminal K with a particular part of speech interpretation and with a particular selection restriction. At the same time, all graphs t1, t2, ... tn interpreted by the same NF expression "K" are substitutable for one another, regardless of whether the root and end nodes of ti are identical or different from those of tj (1 less than or equal to i, j less than or equal to n; i not equal to j).

A. Normal Form Grammar Rules

NF grammar rules have the format of subscript grammar rules. An NF rule consists of a rule name and the rule statement. The antecedent in the rule statement is a complex symbol, a Normal Form expression (NF expression). The terms in the consequent of an NF rule are "complex standard rule names", some of which are complex terminal symbols. The name R of an NF rule is associated with the consequent of an NF rule and uniquely identifies it. If a consequent p of an NF rule is interpreted by more than one NF expression, as in the case of genuine ambiguity, each of the different NF rules will have the same name R.

1. Consequent of NF Rules

The terms in the consequent of an NF rule consist of either a standard subtree, represented by the name of the standard rule which was used to construct the subtree, or by a sequence of standard subtrees (of connected subtrees).

With the rules R1 : V S --> D AUX + V S'

and R2 : V S' --> V PRED + V SUBJ,

the NF consequent terms R1, R2, and R1R2 would represent the subtrees A, B, and C, respectively.

```
A)      V S          B)      V S'          C)      V S
        / \                  / \                   / \
       /   \                /   \                 /   \
      /     \              /     \               /     \
  D AUX     V S'      V PRED     V SUBJ      D AUX     V S'
                                                       / \
                                                      /   \
                                                     /     \
                                                V PRED    V SUBJ
```

The order of the terms in the NF rule consequent represents the order of the nodes in the standard subtree interpreted by the NF rule: top-to-bottom and right-to-left.

2. Conditions and Operations in NF Rules

All conditions and operations of subscript grammar rules can be stated in NF rules. However, it has not been shown to be necessary to state operations between terms in the consequent of an NF rule. Since various conditions can be stated for the terms in the consequent, different NF expressions can be assigned to the same general subtree, if desired, dependent on the conditions stated.

The carry and define operations which assign subscripts and/or values to rule antecedents of the syntactic subscript rules can simply be carried over into the NF expression. E.g., "cousin" would acquire the values 'human' and 'female' in 'His cousin killed herself', the features 'human' and 'male' in 'His cousin killed himself', by means of the rule

```
        V  "COUSIN"                C 13

        $  2.1                     $ TY
```

where

```
    C 13 is        V N             * COUSIN
                   + TY(HU,M,F)
```

3. **NF Rule Antecedents (or Normal Form Expressions)**

An NF expression is a complex symbol consisting of a category symbol identified by the operator V; and, optionally, of--

essential subscript and value symbols identified by the operators $ or +;
selectional subscripts and values identified by the operator A;
a degree operator D.n;
a preference operator P;
the operator N.

The category symbol indicates the NF reading of a standard subtree. This reading can be a series of letters or of digits, a meaningful symbol, or a sequence of meaningful symbols. Essential subscripts represent semantic subscripts and values of NF expressions. The values of selectional subscripts represent standard structures. They are used to select similar standard output structures during translation. The value n of the degree operator gives the number of nonterminal end nodes in the subtree interpreted by the NF expression. The preference operator P permits the selection of one NF expression from two or more which interpret the same syntactic subtree.

In order to facilitate the coding and checking of normal form rules, the operator N was added. Information associated with this operator is not part of the NF expression. Thus, N TM(ARBEITEN) indicates that the NF expression interprets the terminal 'arbeiten'.

Normal form rules are usually divided into two main groups on the basis of whether or not they contain non-terminal or only terminal nodes. A "zero level" transfer (normal form) rule consists solely of terminal constituents, that is, there are no variable constituents within the rule. For example,

```
C 4277         V KINETICS            C 20791
T GER TRAN     A CAT(N)
               N TM(KINETIK)
```

states that German C 20791, which is the dictionary rule for the noun "Kinetik" is given the transfer name KINETICS. This German transfer rule corresponds to the English transfer rule:

```
C 4284         V KINETICS            C 20765
T ENG TRAN     A CAT(N)
```

which realizes KINETICS as the English lexical entry C 20765, in this case the noun "kinetics" with its associated features. (In this example the transfer name is identical to the English surface element, although such is not characteristically the case.)

More than one lexical item may occur in the transfer rule, and the number of elements may be different in the two languages. For example, transfer maps the German tree for the phrase "ins Leben rufen" into the single English word "found" through the following rules:

```
C 4859  V FOUND    C 5031  C 21289  C 54  C 149  C 9000.65  C 4353
T GER   A CAT (V                    A 1   A 1    A 1        A 1
 TRAN        +P)
        N TM(INS                          B 3    B 4        B 5
           +LEBEN
           +RUFEN)
```

```
C 415         V FOUND     C 4405
T ENG TRAN    A CAT(V)
```

(Here the A 1, B 3, B 4, and B 5 designations are an indication of the way in which the nodes are interconnected in the tree. Each term is assumed to branch down from the previous term when no condition is indicated. A 1 means that the branch is at the same level as the previous term and is the second branch at that level. B 3 indicates the third branch, etc.)

German "begruenden" is also mapped into English "found" by the following rule:

```
C 419         V FOUND     C 4243
T GER TRAN    A CAT(V)
              N TM(BEGRUENDEN)
```

Thus the mappings from German to English are not unique. Several German phrases may share a single English translation, while a particular ambiguous German phrase may have more than one English equivalent.

In addition to the transfer rules which consider only terminal nodes, one must consider the non-terminal rules. Since such rules contain one or more internal variables, they are referred to an "non-zero" transfer rules. The format is identical to that of zero level rules, except that the first term contains a designation of the number of unspecified "sons" the node in question carries. For example,

```
C 5229          V X:PREP+NP            C 40115
T GER TRAN      D 2                    $ TYP(PR)
```

transfers a prepositional phrase. If one examines German surface rule C 40115, one finds that it states that an adverb consists of a preposition followed by a noun phrase. The PREP and NP are variable constituents, and their presence is indicated by the D 2 under the transfer rule name.

VII. ENGLISH STANDARD (DEEP) STRUCTURE

The purpose of this Standard Production phase is to associate with the normal form reading of a sentence the standard trees of all English output sentences having the same normal form reading. I.e., the object is to find all the English sentences which have the same meaning as the German sentence. Secondly, it tries to arrange the constituents of such sentences in an order which closely reflects an acceptable English surface order. This phase involves two sub-components, File Entry Construction, and Output Standard Choice.

A. File Entry Construction

The purpose of File Entry Construction is to associate with the normal form readings provided by the analysis of a German input sentence all well-formed English standard trees (dominated by the symbol S) with the same meaning as the input sentence. At the same time, File Entry Construction tries to produce translation sentences whose structural description is similar to the description of the German sentence, in order to reduce the number of translations provided for each input sentence.

File Entry Construction operates with the English transfer grammar and the English standard grammar. It is executed in two phases, Interlingual Mapping, and Synthesis.

1. Interlingual Mapping

The purpose of Interlingual Mapping is to associate with the German normal form expression all English sub-trees interpreted by the corresponding English normal form expressions. At the same time, it checks whether the retrieved standard sub-trees can be connected. Two standard sub-trees can be connected if the label of the root of a sub-tree A is identical to a label of a node of sub-tree B into which A is to be linked. Thus the sub-trees

```
  -----
  | A |          and            N
  -----                        /|
    |                         / |
    |                        /  |
    |                       /   |
    |                   -----   |
 SPECIAL                | A |   N
                        -----
```

can be connected in $\boxed{A}$. The sub-trees

```
        -----
        | A |           and            N
        -----                         /|
          |                          / |
          |                         /  |
          |                        /   |
          |                       /    |
       SPECIAL                   N     N
```

cannot be connected.

File Entry Construction constructs the superficial English standard trees by working from left-to-right and bottom-to-top. It performs, essentially, the following five operations:

- retrieval of the English transfer rules with the same normal form expression;
- performance of subscript/value check (a German normal form expression is interpreted as a rule condition which must be satisfied by the English normal form expression). The English normal form expressions which do not match are deleted;
- retrieval of all English standard sub-trees interpreted by matching English normal form expressions;
- conflation of terminal symbols with identical category symbols;
- connection of the retrieval standard sub-trees with the sub-trees obtained so far (recall the the algorithm works from bottom-to-top) and rejection of those sub-trees which do not connect.

2. Synthesis

The function of the Synthesis portion of File Entry Construction is to check that the superficial standard trees are indeed well-formed according to the standard grammar of the language. Those sub-trees which do not satisfy this condition are deleted. Synthesis also checks that nodes of the remaining trees have the subscripts and values which were stated in the English transfer rules.

Synthesis operates with the English standard rules that have been provided by the interlingual mapping. For each rule it performs essentially four operations:

subscript/value check for each term of the rule,
operations,
left-side construction, and
left-side check.

During left-side check, the constructed left side is compared against the left side stipulated by the English transfer rule. Only those columns of the constructed left side which contain values which match the values provided by transfer are retained. For example, the English transfer rule SING (singular) refers to a sub-tree which builds on a noun and noun endings, and specifies that the result should be the number singular. The result of left-side construction, however, is the subscript NU with the values singular and plural. Left-side check eliminates the column or columns containing the value plural.

The output of Synthesis is a well-formed English standard tree (or set of trees in cases of structural ambiguity of the input text). The conventions used in writing the rules and the structure of workspaces are similar to those used on the German side of the translation system.

VIII. ENGLISH LEXICON

A. English Dictionary Grammar

English lexical items may be found in the English Surface Grammar - Dictionary [EFG-D] and in the English Verb Dictionary [EVD]. The format of English lexical rules is identical with that used for the German lexicon, as can be seen from the following example:

```
C 10577         V A             * FLUORESCENT
T ENG DICT      + CL(01)        P
                + TM(NT,IN)
                + ON(C)
```

Each rule has the surface form on the right with an asterisk (*) to indicate that it is a terminal node. There may be more than one word on the right side, as in:

```
C 6327          V AV            * IN PART
T ENG DICT      /               P
```

Such rules usually have a preference operator to prevent multiple analyses resulting from processing of the individual elements of the entry.

The left side of the dictionary rule contains the category symbol together with semo-syntactic features and operators as appropriate. The operators are identical to those used in the German rules and so need not be discussed here. The feature system for English does differ from the German features (grammatical gender does not occur in English, for example) and so will be described below.

1. English Noun Features (Category Symbol = N)

Nouns are semantically classified and in addition have descriptors indicating the types of attributes which they may take. The subscripts for nouns are:

CL	=	morphological class
ON	=	onset
TY	=	semantic type of noun
OB	=	object (in case of deverbative nouns, as e.g., "dependence on")
TO	=	semantic type of object
TA	=	takes attribute
RL	=	relative adverb (for deverbative nouns)
FM	=	form (for formalized adjectives)
SX	=	sex
IO	=	interpretation of object
FC	=	form of complement
TC	=	semantic type of complement
DF	=	derived form

CL (morphological class) represents the inflectional endings (i.e., plural and possessive forms) possible with the noun stem. The values are numeric, as in German.

Class	Sing.	Plural	Possessive (if unpredictable from inflectional endings)	Examples
01	0	-S		work, altar
02	0	-ES		apparatus, class
03	-E	-ES		chang(e)
04	0		-'S	study, intensity
05	0		-'	optics
06	0	0	-'S	sheep, aircraft
07	0	0	-'	series
08		0	-'S	men
09		0	-'	groats
10		-ES		studi
11	0	-E		stria, alumna
12	0	-TA		stroma
13	-IS	-ES		cris, analys
14	-ON	-A		criteri, automat
15	-UM	-A		dat
16	-US	-I		radi
17	0	-'S		A, B, C . . .

ON (onset) must be specified for English nouns since the onset character conditions the co-occurrence with the indefinite determiner "a/an". The values are thus:

C = consonant
V = vowel

TY (semantic type) is used to represent the broad semantic type associated with the noun, thus allowing semantic co-occurrence restrictions with modifiers and verbs to be tested. The values are as they are for German, namely:

E = entia (anything)
PO = physical object
AB = abstract
NT = non-tangible (a new category primarily for "wave" phenomena, e.g., "light", "sound", "electricity")
AN = animate
PL = plant
IN = inanimate
HU = human
F = female
M = male
AL = animal
NM = proper name
T = title
CO = collective (components may be counted; can be used with the verb "disperse"; e.g., "group", "herd", "government")
BP = body part
MS = mass (homogeneous; may occur without article in the singular; e.g., "milk", "sand")
MA = machine (since they can perform some human activities)
QU = quantity (____ + ("of") NP; e.g., "group", "glass", "half", as in "a glass of milk")
CN = count (abstract countable nouns, e.g., "idea")
UN = unit (ADV = QUANT + ____; e.g., "mile", "year", as in "five miles long", "to wait two years")
TM = time

These values may be used in combinations; e.g., the English noun "government" which has the features TY(HU+CO,AB) indicating both human and collective, or, abstract. This value system may be represented in tree form as shown on the following page.

```
                                   E
                                   |
                                   |
               -----------------------------------------
               |             |            |           |
               |             |            |           |
               NT            PO           QU          AB
                             |                        |
                             |                        |
            -------------------------------    ------------------
            |                   |        |     |   |   |   |   |
            |                   |        |     |   |   |   |   |
            |                   |        |     NM  CN  TM  UN  QU
            |                   |        |
            AN                  PL       IN
            |                   |        |
            |                   |        |
       ------------------   ----------   ----------------------
       |               |    |   |   |    |   |   |   |   |   |
       |               |    |   |   |    |   |   |   |   |   |
       HU              AL   CO  NM  MS   NM  MA  CO  QU  MS  BP
       |               |                         |
       |               |                         |
----------------  -------------                  QU
|  |   |   |  |   |   |   |  |
T  NM  CO  M  F   NM  CO  M  F
```

OB (object) The values for the subscript (if relevant) are all prepositions, spelled out.

TO (semantic type of object) has the values PO, AB, etc., as defined under TY above.

TA (attributive) is used with nouns which take an attribute phrase. The values reflect the syntactic form of the attribute:

- MI = marked infinitive (e.g., "attempt", as in "the attempt to do something")
- CL = main clause used as an attribute of a noun
- TH = "that"-clause (non-relative "that"-clauses; e.g., "his claim that this was so")
- DIR = directional adverbial complement (e.g., "a trip across Europe")

RL (relative adverb) is used with nouns which characteristically take a relative adverb phrase as a modifier. The feature is not usually necessary but may be useful in certain kinds of texts and so is included in the general scheme. The values are:

WHERE	= where (e.g., "the place where I saw you")
WHERETO	= whereto (e.g., "the town where you went")
WHY	= why (e.g., "the reason why he did it")
WHETHER	= whether (e.g., "the question whether this is so")
HOW	= how (e.g., "his knowing how it was done")
WHEN	= when (e.g., "the time when I lived there")

FM (formalized adjective) indicates derived nouns, such as adjective forms used as nouns. The values used indicate the syntactic class of the form in question:

A = adjective
I = infinitive
G = gerund

SX (sex) is used to indicate the natural gender associated with a noun. Although English, unlike German, lacks grammatical gender, natural gender is necessary for pronoun agreement. Absence of the subscript may be taken to represent "neuter" gender. The subscript SX has two possible values:

FE = female
MA = male

IO (interpretation of object) is used to interpret the potential objects which a deverbative nominal may take:

O = first object
O2 = second object
O4 = reflexive object
LA = lambda (null)

FC (form of complement) is used primarily with deverbative nominals which may take a complement of their own. This complement is usually a noun or a phrase representing the subject or object of the verb from which the noun is derived. Since most nominal complements are prepositional phrases, the values of FC are usually prepositions such as WITH, IN, OF, etc. The value lambda (LA) is used to indicate absence of a preposition in the complement phrase.

TC (type of complement) is the semantic type of a complement used with a deverbative nominal. The values are those of TY (type) as given above.

DF (derived form) is often used with derived nominals, usually deverbative forms such as gerunds and agentive nouns, but also sometimes with nouns derived from adjective stems. It further specifies the form underlying the nominal. The values for DF are:

VI = intransitive verb
VT = transitive verb
VR = reflexive verb
A = adjective

2. English Determiner Features (Category Symbol = DET)

English determiners include all forms which occur as prenominal determiners. Thus forms such as "a", "the", "that", "some", "each", "her", etc., are included. (See the latter portion of the section on German determiners for a discussion of the overlap between determiners and pronouns.) Subscripts used with determiners are:

NU = number
RO = onset charac er
FM = grammatical form
WD = surface word

NU (number) is as one would expect:

S = singular
P = plural

RO (required onset) is used with the forms "a" and "an", which must agree with the initial letter of the following word:

V = vowel
C = consonant

FM (grammatical form) is used to indicate the syntactic function of the determiner. The values are:

DET = determiner
DEM+P = demonstrative pronoun
REL+P = relative pronoun
IND+P = indefinite pronoun
POSS+P = possessive pronoun

WD (surface word) is sometimes used when it is necessary to specify the particular determiner in other rules. The value is the actual surface form represented by the dictionary rule, for example:

SUCH

3. English Pronoun Features (Category Symbols = PRN or REFL)

English pronouns classified under the category symbol PRN are personal pronouns, while REFL is used for reflexives.

The following subscripts are used with pronouns:

NU = number
PS = person
FM = grammatical form
TY = semantic type
CA = grammatical case (function)
OR = object function
TM = surface realization
T = interrogative type
FE = file entry

NU (number) is used to indicate number distinctions. Two values are possible:

S = singular
P = plural

PS (person) is expressed in English pronouns through three values:

1 = first person
2 = second person
3 = third person

FM (grammatical form) is an indication of the syntactic use of the pronoun. The values are thus:

IND+P = indefinite pronoun
PERS+P = personal pronoun
REF+P = reflexive pronoun
REL+P = relative pronoun
INT+P = interrogative pronoun

TY (semantic type) indicates the semantic type of the referent. The values are those found in the semantic TYPE tree for nouns, q.v.

CA (grammatical case) is similar in function to case in German in that it indicates the syntactic function of the pronoun. The values used in English are:

N = nominative (subject function)
O = objective (object function)

OR (object function) indicates the syntactic function of the pronoun when used as an object. The values are the same as those used in enumerating objects which may occur with verbs, namely:

O = direct object
O2 = indirect object
O3 = potential special object
O4 = reflexive object
LA = unspecified

TM (surface realization) is used in actualizing dummy pronouns as surface forms, in particular the dummy reflexive. The values are the actual surface realizations of the dummy.

MYSELF
YOURSELF
HIMSELF
HERSELF
ITSELF
OURSELVES
YOURSELVES
THEMSELVES
LA = unspecified

T (interrogative type) is used with interrogative pronouns and indicates the semantic type of the expected answer. The values are those associated with TY for nouns, q.v.

FE (file entry) is a subscript created during file entry construction. The values, which are numerical, code for the surface realization of the node in question.

4. English Adjective Features (Category Symbol = ADJ)

In the form of the lexicon in use at the time of the LRC Demonstration (June, 1974) adjectives were given one or more of the following subscripts. (For a description of the revision of the system of adjective features which was in progress at that time, cf. the LRC report of Aug., 1973.)

CL = morphological class
TY = type of adjective
FM = form of adjective
MD = modifies nouns of the specified type
RA = requires an adverb
OB = form of object
TO = semantic type of object

CL (morphological class) uses two-digit numbers to specify each unique set of paradigmatic forms for the three degrees of adjectives and any adverb built on the same stem. Column 2 is a summary of the type of paradigmatic material utilized by that class. As elsewhere, a blank slot signifies the non-occurrence of a form, and a 0 indicates that the stem takes a zero-affix.

Class		Pos.	Comp.	Sup.	Adv.	Example stems
01	{RR*}	0	more	most	-ly	lateral
02	{P1,C2}	0	more	most		Greek
03	{P1,C2,A2}	0	more	most	-ally	photographic
04	{P1}	0				ready
05	{P2,C2,A3}	-e	more	most	-y	capabl
06	{P1,C2,A4}	0	more	most	0	baby-like
07	{C1,A1}		-er	-est	-ly	readi
08	{A1}				-ly	benedictori
09	{RG**}	0	-er	-est	-ly	vast
10	{P2,C1,A5}	-e	-er	-est	-ely	clos
11	{P1,A4}	0	-ther	-thest	0	far
12	{P2,C1,A3}	-e	-er	-est	-y	simpl, abl
13	{P1,C1}	0	-er	-est		few
14	{P1,C1,A4}	0	-er	-est	0	low
15	{P1,C1,A3}	0	-er	-est	-y	full
16	{P2,A3}	-e			-y	singl
17	{P2,C1}	-e	-er	-est		whit
18	{P1,C3,A1}	0	-ger	-gest	-ly	snug
19	{P1,C4,A1}	0	-mer	-mest	-ly	dim
20	{P1,C5,A1}	0	-ner	-nest	-ly	thin

CL cont'd

Class		Pos.	Comp.	Sup.	Adv.	Example stems
21	{P1,C6,A1}	Ø	-ber	-best	-ly	glib, drab
22	{P1,C7,A1}	Ø	-ter	-test	-ly	hot
23	{P1,C8}	Ø	-der	-dest		red
24	{P1,C5}	Ø	-ner	-nest		tan
25	{P1,C7}	Ø	-ter	-test		fat
26	{P1,C3}	Ø	-ger	-gest		big
27	{P1,C4}	Ø	-mer	-mest		trim

* regular, Romance
** regular, Germanic

TY (type of adjective)

MSR = measurable (e.g., "wide" or "strong" as in "five inches wide", "seven men strong")
TM = the adjective may undergo "tough movement" (e.g., "hard", "easy")

FM (form of adjective)

PRPL = the adjective is in the form of a present participle
PAPL = past participle

MD (type of noun modified) has all the semantic categories of noun as values, plus--

TH = "that"-clause
PLU = plural, mass, or collective noun

RA (requires adverb) has as possible values those given for the subscript RA for verbs, q.v.

OB (form of object)

GG = genitive
DG = dative
AG = accusative
All prepositions, spelled out

TO (semantic type of noun) uses the values of TY for nouns, q.v.

5. English Verb Features (Category Symbol = V)

Each English verb is marked by some or all of the following subscripts.

CL = morphological class
TY = type of verb
TS = semantic type of subject
FS = syntactic form of subject
OB = syntactic form of object(s) or complement(s)
TO = semantic type of object
RA = required adverbials
OA = optional adverbials

CL (morphological class of the stem)

Class	Base	3rd Sing.	Present Partic.	Past	Past Partic.	Example
01	-E	-ES	-ING	-ED	-ED	revolv
02	0	-ES	-ING	-ED	-ED	reach, miss
03	0	-S	-ING	-ED	-ED	soar
04	0	-S	-BING	-BED	-BED	rub
05	0	-S	-DING	-DED	-DED	stud
06	0	-S	-GING	-GED	-GED	plug
07	0	-S	-KING	-KED	-KED	panic, frolic
08	0	-S	-LING	-LED	-LED	pal
09	0	-S	-MING	-MED	-MED	stem
10	0	-S	-NING	-NED	-NED	don
11	0	-S	-PING	-PED	-PED	stop
12	0	-S	-RING	-RED	-RED	blur
13	0	-SES	-SING	-SED	-SED	gas
14	0	-S	-TING	-TED	-TED	knit
15	0	-ZES	-ZING	-ZED	-ZED	quiz
16	0	-S	-ING	-D	-D	agree
17	0	-S	-ING	-ED	-ED	show
18	0	-S	-ING	0	0	read
19	0	-S	-ING			dream
20	0	-ES	-ING			focus
21	-E	-ES	-ING			mak
22				0	-N	wove
23				-E	-N	swor
24				0	0	unwound
25	0	-S	-ING		-N	see
26	0	-S	-ING		-EN	eat

CL cont'd

Class	Base	3rd Sing.	Present Partic.	Past	Past Partic.	Example
27	-E	-ES	-ING		-EN	giv
28				0		fell, ate
29	0	-S		0	0	cut
30			-ING			cutt
31	0		-ING			cry, imply
32		-ES		-ED	-ED	cri, impl
33					0	lain

TY (type of verb) relates to transitivity

- VT = takes at least one object which is not a reflexive pronoun
- VTC = takes a cognate object only; we define a cognate object as the true cognate and all nouns subsumed under that term, as e.g., "to dance a rain dance" or "...a waltz".
- VR = takes an object which MUST be reflexive
- VT,VR = takes at least two objects, one of which must be reflexive and one which is not reflexive
- VI = intransitive
- NP = the verb does not passivize; verbs marked VI or VR do not need this descriptor.
- NG = the verb does not form the progressive.

TS (type of subject) the values associated with this subscript are the usual semantic categories of nouns, q.v. In addition, the value

P = plural noun only

may be used to describe the subject a verb requires.

FS (syntactic form of subject) [This subscript is omitted if the verb allows only a noun phrase as subject.]

- NP = noun phrase
- IT = "it"
- TH = "that"-clause
- MI = marked infinitive
- FT = "for-to" complement
- GR = gerund
- ICL = interrogative clause
- IMI = interrogative adverb + unmarked infinitive

OB (object or complement syntax)

O = noun phrase (NP) as object
CL = main (subjunctive) clause
PAPL = past participle
I = unmarked infinitive
BC = takes "be" + NP or ADJ
CM = takes optional "be" + NP or ADJ ("think")
NC = takes NP complement without "be" ("elect")
NA = takes NP or ADJ complement without "be"
AC = takes ADJ complement without "be"
TH, MI, etc. = as defined above for FS
all prepositions, spelled out

TO (type of object) These values those of TY for nouns, plus:

P = plural noun only
R = reflexive
RCC = reciprocal

RA (required adverbials)

PLC = place (locative OR directional)
DIR = direction to
ORN = origin (direction from)
TIM = time (punctual OR durational)
PNC = punctual
DUR = durational
MAN = manner
MSR = measure
AC = adjective complement (for sensory verbs, e.g., "smell good")

OA (optional adverbials) is always associated with the same value:

DOR = direction or origin (adverb of directionality)

6. English Modal Features (Category Symbols = FIN or DO)

This category is for verbs which are used as auxiliaries in English. That is, they may be followed by a non-finite form of another verb, forming a complex predicate. Unlike the verbs with the category symbol V, the FIN's are included in their fully inflected forms in the lexicon.

Also in the dictionary one finds modal verbs under the category symbol DO. These forms do not occur in the German surface structures but are sometimes necessary for fluent English translation. Forms of "do" are included as well as "may" and "would" when used as subjunctive markers.

The subscripts used for modals are similar to those of verbs:

PS = person
NU = number
TN = tense
MD = mood
VC = voice
FM = form
RQ = required non-finite form
WD = surface word
TY = type
FS = form of subject
TS = type of subject
IS = interpretation of subject
FO = form of object
TO = type of object
IO = interpretation of object
TM = surface form
FE = file entry

PS (person) specifies the distinctions of grammatical person.

1 = first person
2 = second person
3 = third person
LA = lambda (unmarked)

NU (number), as one might expect, has as possible values:

S = singular
P = plural
LA = lambda (unmarked)

TN (tense) indicates the tense of the verb form. Since English has two inflectional sets of suffixes for tense, there are two possible values (in addition to the null value):

PR = present
PA = past
LA = lambda (unmarked)

MD (mood) for English has the associated values:

I = indicative
S = subjunctive
IR = irrealis
LA = lambda (unmarked)

VC (voice) indicates whether the modal serves as finite form in active or in passive sentences. The values are thus:

A = active
P = passive

FM (form of modal) is used with non-finite forms of the modal verbs as an indication of grammatical usage. The values are:

I = infinitive
PAPL = past participle
F = finite
GR = gerund
N = (used with "may" and "would" as subjunctives)

WD (word group) is introduced to specify the stem form of the modal in question, since the actual orthographic representations may vary considerably with changes in tense, mood, person, and number. The values are:

BE = be
BEC = become
C = can
H = have
M = may
REM = remain
SH = shall
WT = want
W = will

TY (type of modal) indicates potential syntactic usage.

M = modal
C = copula
A = auxiliary
BT = (used with existential use of BE)
V = verb (i.e., as the main verb of the clause)

FS (form of subject) indicates the syntactic form required as subject of the modal. The values are:

S = subject
TH = clause

TS (type of subject) designates the semantic class of potential subjects. Usually the values are:

E = entia (everything)
TH = that clause

IS (interpretation of subject) only occurs with the value:

S = subject

FO (form of object) marks the syntactic class of elements which could be used as objects. The values are:

O = object
LA = lambda

TO (type of object) enumerates the semantic category of the object. Since this is not constrained with modals, the values are:

E = entia
LA = lambda

IO (interpretation of object) suggests the function of any objects which may occur. The values are:

O = direct object
LA = lambda

TM (surface form) is used with dummy modals, just as it is in transfer rules, as a means for indicating the surface realization. This is more an aid to the linguist in writing and checking rules than it is a feature used to ensure proper translation by the system.

FE (file entry) is a subscript created during file entry construction. The values are numerical and they code the surface realization of the node in question.

8. English Preposition Features (Category Symbol = PREP)

TY = semantic type of preposition
RC = requires complement (noun phrase or adverb)
TC = semantic type of complement
POS = position (pre- or post-posed)

TY (type) Cf. TY values for adverbs, below.

RC (requires complement)

NP = noun phrase
AV = adverb

TC (semantic type of complement)

all values given for the subscript TY of nouns, q.v.
all values given for the subscript TY of adverbs, q.v.

POS (position)

PRE = pre-posed to the NP or AV
POST = post-posed to the NP or AV

7. English Adverb Features (Category Symbol = AV)

One-word adverbs (including those derived from adjectives and present and past participles) are given some or all of the following subscripts. TY and MD are mandatory.

- TY = semantic type of adverb
- PA = paraphrasability (relevant only for parenthetical adverbs)
- MD = modifies (the adverb may modify verbs, sentences, or NP's)
- TS = semantic type of sentence subject required (relevant only with adverbs modifying verbs and, possibly, sentences)
- TV = semantic type of verb with which the adverb may be used (relevant only with adverbs modifying verbs)
- POS = position (pre- or post-posed; sentence initial, medial, or final)
- RC = requires complement (adverbs, clauses or phrases)
- OC = optional complement
- TN = tense (the adverb requires that the verb occur in a specific tense(s); this subscript is not coded if the same information is contained under TY in one of the values PR, PA or FU)

Each of the subscripts in the list above is associated with one or more values describing the characteristics of the particular item being classified or its selection restrictions. Some values may carry further, more precise specification.

TY (semantic type) used with adverbs, prepositions, and conjunctions.

- P = parenthetical
- DEF = definite
- IND = indefinite
- L = location, which may be specified as
  - STA = static
  - DI-T = direction to
  - DI-F = direction from
- T = time, which may be specified as
  - PR = present
  - PA = past
  - FU = future
  - PR-T = prior to
  - SIM = simultaneous with
  - PO-T = posterior to
  - PU = punctual
  - DU = duration (time span answering the question "how long?", e.g., "FOR eleven days")

TY cont'd

FR = frequency (repetitive)
SE = sequential (SE without INC or TRM means "sequential but not initial or final", e.g., "secondly")
INC = incipient
TRM = terminating
INST = instantaneous (point in time, e.g., "AT 8 P.M.")
EXT = extended (time span answering the question 'when?', e.g., "today")

M = modal, which may be specified as

MAN = manner
SM = state of mind
EV = evaluation of subject (It is ADJ of SUBJECT to INFINITIVE: He wisely did it - It was wise of him to do it.)
COM = comparison
COM-PEJ = comparison pejorative
RES = restrictive
MOD = mode of existence

D = degree, which may be specified as

LS = lower scale
MED = medium
H-S = higher scale
APP = approximation
COM = comparison

CA = cause
PP = purpose
I = instrumental
R = result
CD = condition
CC = concessive
MO = modality
ME = measure
S = sociative
A = adversativity

PA (parenthetical) pertains to paraphrasability.

I = "it" - "that" paraphrase possible: "He will probably come" - "It is probable that he will come".
W = post-sentential which-relative clause possible: "He surprisingly works slowly" - "He works so slowly that it is surprising".
H = "it" - "how" paraphrase possible: "He works surprisingly slowly" - "It is surprising how slowly he works".

MD (modifies)

S = sentence
D = declarative
Q = question
I = imperative
N = negated D, Q, I, or S
V = verb
NP = noun phrase
AV = adverb (including PRPH)
NU = numbers
AJ = adjective
E = equative
P = positive
C = comparative
SP = superlative

TS (type of subject)

P = plural (i.e., the adverb requires a plural subject or a singular subject with a "with"-phrase)

POS (position)

A = ante (= pre-posed) P = post (= post-posed)	relevant for modifiers of NP, A, AV, or NU only
I = sentence-initial M = sentence-medial F = sentence-final	relevant for modifiers of S and V only

RC (requires complement)
OC (optional complement)

All prepositions, spelled out
AV = any type of verb
AV-PLC = adverb of place
AV-TIM = " " time
AV-PNC = " " punctuality
AV-DUR = " " duration
AV-LOC = " " location
AV-DIR = " " direction to
AV-ORN = " " origin (direction from)
AV-MAN = " " manner
AV-MSR = " " measure

OC cont'd

AC	= adjective complement
TH	= "that"-clause
MI	= marked infinitive
FT	= "for-to" complement
GR	= gerund
ICL	= interrogative clause
IMI	= interrogative adverb + marked infinitive
GG	= genitive
DG	= dative
AG	= accusative
NPG	= noun phrase

TN (tense)

PR = occurs with verbs in present tense
PA = occurs with verbs in past tense
FU = occurs with verbs in future
PF = occurs with verbs in any perfect tense (PF may also be used together with PR, PA or FU to indicate present perfect tense, etc., as relevant)

Some values may be used in combinations, as indicated below.

Value Trees:	Possible Value Combinations:

```
      any type
        / \
       /   \
      /     \
    DEF     IND
```

Any value of TY may be combined with DEF or IND as relevant.

```
             L
            /|\
           / | \
          /  |  \
         /   |   \
        /    |    \
       /     |     \
      /      |      \
     /       |       \
   STA     DI-T      DI-F

where?  whereto?  from where?
```

L may be combined with lower nodes (e.g., L STA).

```
          T
         /|\
        / | \
       /  |  \
      /   |   \
     PR   PA   FU
or:

          T
         /|\
        / | \
       /  |  \
      /   |   \
     /    |    \
   PR-T  SIM  PO-T
```

Combine T with any of the 6 lower nodes. However, time adverbs which do not specifically indicate past, present, or future do not get the values PR, PA, or FU.

```
                                                 Possible
     Value Trees (Cont'd):                  Value Combinations:
     ---------------------                  ------------------

                 any sub-class              Combine any of the 6
                     of T                   T-dominated nodes above
                    /|\                     with the lower nodes
                   / | \                    in this tree, as
                  /  |  \                   relevant.
                 /   |   \
                /    |    \
               /     |     \
              /      |      \
             /       |       \
            /        |        \
           /         |         \
when ?   PU          DU   how    FR
         /|\         / \  long ?
        / | \       /   \                    Use DU with DEF or IND
 INST    SE   EXT  INC    TRM                to mean 'limited' or
         /\      since    until              'unlimited duration',
        /  \     when ?   when ?             respectively.
      INC   TRM


                      M                      Combine M with lower
                     /|\                     nodes.
                    / | \
                   /  |  \
                  /   |   \
                 /    |    \
                /     |     \
               /      |      \
              /       |       \
             /        |        \
how ?    MAN         COM         RES
         / \          |
        /   \         |        with respect
       /     \        |         to what ?
     SM       EV     PEJ

                like who ?
                like what ?
```

Value Trees (Cont'd):

Possible Value Combinations:

```
                     D
                     |
                     |
      ----------------------------------
      |       |       |       |       |
      |       |       |       |       |
      |       |       |       |       |
     L-S     MED     H-S     APP     COM
```

to what degree ?

Combine D with lower nodes.

```
                     S
                     |
                     |
      ----------------------------------
      |      |      |      |      |      |
      |      |      |      |      |      |
      |      |      |      |      |      |
      D     D N     Q     Q N     I     I N
```

Combine S or A with any of the lower nodes as relevant.

S by itself means that the particular adverb can modify any of the six types of S; the same is true of A.

```
                     A
                     |
                     |
                     |
             -----------------
             |     |     |     |
             |     |     |     |
             |     |     |     |
             P     E     C     SP
```

9. English Conjunction Features (Category Symbol = CONJ)

C-I = connects or introduces (clauses, noun phrases, verb phrases, etc.)
TY = type of conjunction

C-I (connects or introduces)

MC = main clause
SC = subordinate clause
A = adjective or adverb (phrase)
N = noun (phrase)
V = verb (phrase)

TY (type)

CONJ = conjunctive (takes pl. verb: 'and')
DISJ = disjunctive (takes sg. verb: 'or')

In addition, all semantic features under the subscript TY of adverbs, above, may be used for conjunctions.

IX. ENGLISH SURFACE STRUCTURE

A. Output Syntactic Choice

The purpose of Output Choice is basically twofold. It selects the values of lexical dummies that need to be printed by lexical spellout, and it determines the order in which the constituents are to be printed out. The macros of the English Standard Choice grammar and the Choice instructions contained in the English Standard rules are both used by Output Syntactic Choice.

The Output Standard Choice algorithm is nearly identical to the first two subcomponents of the Syntactic Choice algorithm, with the following differences. It can accept a workspace in parallel format, and it does not destroy any nodes. (English Standard dummy terms, all of which are rewritten as [DUMMY], are not printed by the Lexical Spellout algorithm.) It can thus perform all operations of Syntactic Choice.

X. SENTENCE PRODUCTION

A. Lexical Spellout

The Lexical Spellout component produces English written text from the output of Syntactic Choice. If the text contains multiple translations for lexical pieces, they are printed out in a vertical format. Lexical elements among which a co-occurrence relation holds receive the same numerical subscript. Lexical Spellout does not print DUMMY terminals nor zero morphemes. If an ending is to be printed which has the operator L (for 'letter') attached, the last letter of the preceding terminal is repeated. Thus, forms like "bigger", "stopped", and others are generated.

The multiple translations which may be provided are a sub-set selected by the algorithm of the translations listed for a term in the dictionary. The multiple translations are all equally good from the system's point of view. The system in effect says 'I am providing you with multiple translations because either the dictionary does not have a finer classification for me to make a decision, or there is not enough disambiguation information contained in this sentence to permit me to make a finer selection'. Obviously, the output format could be changed to show multiple translations in the margin or as footnotes. It seems best, however, not to suppress such translations because they might help the user in attaining a correct interpretation of the meaning of the sentence.

APPENDIX: Area-of-Provenience Classification Tags

One of the methods undertaken in the enrichment of LRC lexical files has been the introduction or addition of classifiers, or "tags", marking entries as to their area of major provenience. (The "tags" are the shortened forms of the classifiers used in the dictionary rules, and are indicated below by being in capital letters.)

The primary sources for this information have been Wildhagen's English-German, German-English Dictionary, and Langenscheidts Enzyklopaedisches Woerterbuch. As might be expected, the systems of classification in the two dictionaries differ. Moreover, linguists engaged in coding entries have occasionally introduced classifiers which their work has shown to be useful. In the following list the classifiers from Wildhagen appear at the margin, and the corresponding Langenscheidt or (in square brackets) linguists' classifiers follow an equals-sign. Ultimately a single classification will be adopted and integrated into all lexical files used in the system.

AEROnautics = AERonautics
AESTHetics
AGRiculture = same
... = ALCHEMY
ANATomy = MEDical, PHYSIOLogy
ANTiquity = ANTIQuity
ANTHRopology
ARCHitecture = same
ARCHAEOLogy
ARITHmetic = MATHematics
ARTS, fine
ASTRonomy = same
ASTROLogy = ASTRology
ATHLetics
BACTeriology = MEDical
BAKing
BIBlical = BIBLical
... = BILLIARDS
BIOlogy = same
... = BIOCHEMISTRY
BOOKBinding
BOTany = same
BOXing
CALLigraphy
... = [CARCinology]
CARPentry
CHEMistry = same

CHRONology
COMmercial = ECONomics
CRICKet
CRIMinal
CRYSTALography = CHEMistry, PHYSics
CULinary
CYCLing
DENTistry = MEDical
DIPLomacy
ECclesiastical = RELIGion
ECONomics = same
ELectricity = ELECTRicity
EMBRoidery
ENGINeering = TECHnology
ENGRaving
ENTOMology = ZOOlogy [ENTomology]
ETHNology = SOCiology
EXCHange, stock = ECONomics
FENCing
FORestry = AGRiculture, GARTENBAU
FORTification = MILitary, ARCHitecture
... = [GAME]
FooTBall
GARDening = GARTENBAU
GENEALology
GEOgraphy = GEOGRaphy
GEOLogy = same
GEOMetry = MATHematics
GRAMmar = LINGuistics
GYMnastics
GYNaecology = MEDical
HERaldry = same
HISTory = same
HOCKey
HORSemanship
HORTiculture = GARTENBAU
HUNTing
ICHthyology = ZOOlogy
INDustry
JURisprudence = same
... = [LANGuage name]
... = LINGuistics
LITerature
LITERary
LOGic = PHILOSophy
MACHinery = TECHnology
... = [MALacology]
MARine = MARitime
MATHematics = same

MECHan·· = TECHnology
MEDici = same
METallurgy = TECHnology
METEORology
... = METRics
MILitary = same
MINing = BERGBAU
MINeRalogy = MINing
... = [Marine InVertibrate Zoology]
MOToring
MOUNTaineering
MUSic(al) = same
MYTHology
NAUTical = MARitime
NAVal = MARitime
NUMismatics
OPTics = PHYSics
ORNithology = ZOOlogy
PAINTing
PALeontology
PARLiamentary
PATHology = MEDical
... = PEDAGOGY
PHARMacology = BIOCHEMISTRY, MEDical
PHILology = LINGuistics
PHILOSophy = same
PHONetics = LINGuistics
PHOTography = same
PHYSics = same
PHYSIOLogy = same
POETry, -ical = same
POLitics = same
PRAEHistory
PRINTing = same
PROSody = METRics
PROVerb
PSYCHology = same
RAILway
RELigion = RELIGion
RHETorical
RUGby
SCIENTific
SCULPture
SKIing
... = SOCIOLOGY
SPINning
SPORT
STATistics = MATHematics
STock EXCHange (dupl.) = ECONomics

SURGery = MEDical
SURVeying
SWIMming
TAILoring
TECHnological = same
TELeGraphy
TELePHony
lawn TENnis
THEATre
THEOLogy = RELIGion
TYPography
VETerinary = same
WEAVing
WIReless = RADIO
ZOOlogy = ZOology

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.